Data-Adaptive Multivariate Density Estimation Using Regular Pavings, With Applications to Simulation-Intensive Inference

A thesis submitted in partial fulfilment of the requirements for the Degree of
Master of Science in Statistics

by Jennifer Harlow

under the supervision of
Dr. Raazesh Sainudiin
and
Dr. Dominic Lee and Dr. Sharyn Goldstien

Department of Mathematics and Statistics
University of Canterbury

2013

[This page intentionally left blank]

Abstract

A regular paving (RP) is a finite succession of bisections that partitions a multidimensional box into sub-boxes using a binary tree-based data structure, with the restriction that an existing sub-box in the partition may only be bisected on its first widest side. Mapping a real value to each element of the partition gives a real-mapped regular paving (RMRP) that can be used to represent a piecewise-constant function density estimate on a multidimensional domain. The RP structure allows real arithmetic to be extended to density estimates represented as RMRPs. Other operations such as computing marginal and conditional functions can also be carried out very efficiently by exploiting these arithmetical properties and the binary tree structure.

The purpose of this thesis is to explore the potential for density estimation using RPs. The thesis is structured in three parts. The first part formalises the operational properties of RP-structured density estimates. The next part considers methods for creating a suitable RP partition for an RMRP-structured density estimate. The advantages and disadvantages of a Markov chain Monte Carlo algorithm, already developed, are investigated and this is extended to include a semi-automatic method for heuristic diagnosis of convergence of the chain. An alternative method is also proposed that uses an RMRP to approximate a kernel density estimate. RMRP density estimates are not differentiable and have slower convergence rates than good multivariate kernel density estimators. The advantages of an RMRP density estimate relate to its operational properties. The final part of this thesis describes a new approach to Bayesian inference for complex models with intractable likelihood functions that exploits these operational properties.

The semi-automatic convergence diagnosis method provides a useful improvement to the Markov chain Monte Carlo RP partitioning method, which is shown to work well in low dimensions. However, the very large data sets required to achieve reasonable estimation errors with four or five-dimensional data can result in extremely long running times. A more flexible prior could mitigate these drawbacks. The kernel density estimate approximation method developed in this thesis may provide the best means of obtaining an RMRP density estimate for small data sets, and for higher dimensional data, if it is used to approximate a suitable kernel density.

The new approach to Bayesian inference for complex models with intractable likelihoods discussed in the final part of this thesis is shown to have considerable potential. The next step should be to carry out further testing of the method in conjunction with RMRP kernel density estimate approximations.

[This page intentionally left blank]

Acknowledgements

I would not have started this research project without the encouragement of my senior supervisor, Dr. Raazesh Sainudiin. I would not have completed it without his continued support, and that of Dr. Dominic Lee, one of my assistant supervisors. They never let me give up, and I have done much more under their guidance than I ever thought myself capable of.

An earlier research project, funded through Dr. Sharyn Goldstien, enabled me to build the genetic models used in Chapter 9. Theses models were based on code originally written by Brendan Bycroft.

My work on regular pavings as structures for organising data and making density estimates started when I assisted Dr. Gloria Teng by implementing some of the algorithms for her Ph.D. thesis. The theory of both the regular paving Markov chain Monte Carlo algorithm and the randomised priority queue algorithm used in this thesis was developed by Dr. Sainudiin and Dr. Teng. Without the experience gained while assisting Dr. Teng, and without the necessity to improve my implementations to meet the computational demands of her research, I would not have already developed and tested some of the extensive code-base that I have used and extended for this thesis.

Dr. Sainudiin had the original idea of arithmetic on mapped regular pavings and supervised me when I undertook an earlier research project developing some of the algorithms and code for these structures. The formal description of most of the algorithms given in this thesis is extensively based on the improvements made by Dr. Sainudiin to the informal efforts that I originally prepared for our joint publications.

Dr. Xibin Zhang and Professor Maxwell L. King of Monash University kindly supplied me with `C` code for their kernel density estimation algorithm, and with permission to convert this to `C++` for use in this research.

I owe thanks to all the other postgraduate students and staff of the Department of Mathematics and Statistics at the University of Canterbury. My especial thanks are due to Steve Gourdie, ICTS Manager for the department. He met my every demand for more memory and faster machines, but — even more valuable — he gave me his friendship. Dr. Jennifer Brown, Irene David, and Hilary Seddon have helped me immeasurably with their calm advice and encouragement.

This thesis was undertaken with the financial support of a scholarship from the University of Canterbury.

[This page intentionally left blank]

Contents

Notation

General symbols

$\mathbb{N}$	the set of all natural numbers
$\mathbb{Z}$	the set of all integers
$\mathbb{R}^d$	the set of all real numbers in dimension d
x	punctual scalar or vector in $\mathbb{R}^d$
A	a set
$\mid A \mid$	number of points in set A
$\mathbb{1}_A(x)$	indicator function: 1 if $x \in A$; 0 otherwise

General statistics and probability

X	random variable
$P(A)$	probability of event A
F or F_X or $F(x)$	cumulative distribution function
f or f_X or $f(x)$	probability density or mass function
$X \sim F$	X has distribution F
$X \sim f$	X has density f
$X_1, \ldots, X_n \overset{\text{iid}}{\sim} F$	independent and identically distributed sample of size n from F
$X_1, \ldots, X_n \overset{\text{iid}}{\sim} f$	independent and identically distributed sample of size n from probability density function f
$f_{X,Y}$ or $f(x, y)$	joint probability density or mass function; joint probability density of X and Y
$f_{X\mid Y}$ or $f(x \mid y)$	conditional probability density; probability density of X conditional on Y
$\Theta \in \mathbb{R}^d$	parameter space
$\theta \in \Theta$	scalar or vector in the parameter space

Models and Estimation

$\mathfrak{F}$	statistical model — a set of distribution functions or density functions
$\widehat{\theta}$	estimate of parameter or vector of parameters
$\hat{f}$	estimate of probability density function f
$\hat{f}_n$	estimate of probability density function f based on n observations

$x^\star$	observed data, the outcome of an unknown model
$t^\star$	a scalar or vector summary of observed data
t	a scalar or vector summary of an outcome from a model

Likelihood

$f\left(x;\, \theta^\star\right)$	probability density or mass function for $X \sim f\left(x;\, \theta = \theta^\star\right)$
$\mathcal{L}(\theta) := f\left(x^\star;\, \theta\right)$	likelihood function for parameter θ given data $x^\star$
$\mathcal{L}_n(\theta) := f\left(x_1^\star, \ldots, x_n^\star;\, \theta\right)$	likelihood function for parameter θ given n observations $x_1^\star, \ldots, x_n^\star$

Bayesian statistics

$f(\theta)$	prior density for θ
$f\left(\theta \,|\, x^\star\right)$	posterior density of θ given data $x^\star$

Intervals and interval vectors

$\mathbb{IR}^d$	the set of all interval real numbers in dimension d
$\underline{\boldsymbol{x}}$	lower bound of $\boldsymbol{x} \in \mathbb{IR}$
$\overline{\boldsymbol{x}}$	upper bound of $\boldsymbol{x} \in \mathbb{IR}$
ι	the first coordinate of maximum width of a box $\boldsymbol{x}$
$\mathrm{wid}\left(\boldsymbol{x}\right)$	the width of an interval $\boldsymbol{x}$
$\mathrm{mid}\left(\boldsymbol{x}\right)$	the midpoint of an interval $\boldsymbol{x}$
$\mathrm{vol}\left(\boldsymbol{x}\right)$	the volume of an interval vector or scalar $\boldsymbol{x}$
$\square\boldsymbol{x}$	interval or box hull of a box $\boldsymbol{x}$
$\mathbb{1}_{\boldsymbol{x}}(x)$	indicator function: 1 if $x \in \boldsymbol{x}$; 0 otherwise

Regular pavings

ρ	root node of a regular paving (RP)
$\rho\mathsf{v}$	node of an RP with root ρ
$\rho\mathsf{vL}$	left child-node of $\rho\mathsf{v}$
$\rho\mathsf{vR}$	right child-node of $\rho\mathsf{v}$
$\boldsymbol{x}_{\rho\mathsf{v}}$	box associated with a node $\rho\mathsf{v}$
$\mathsf{d}_{\rho\mathsf{v}}$	depth of node $\rho\mathsf{v}$ in an RP tree
s	an RP
$\mathbb{V}(s)$	the set of all nodes of an RP
$\mathbb{L}(s)$	the set of all leaf nodes of an RP

$\mathbb{C}(s)$	the set of all cherry nodes of an RP
$\boldsymbol{x}_{\mathbb{V}(s)}$	the set of boxes of all nodes in an RP s
$\boldsymbol{x}_{\mathbb{L}(s)}$	the partition of a root box $\boldsymbol{x}$ described by RP s
$\mathbb{L}^{\triangledown}(s)$	the set of all splittable leaf nodes of an RP s
$\mathbb{S}_{i:j}$	the set of all RPs with a given root box and k splits where $k \in \{i, i+1, \ldots, j\}$
$\mathbb{S}_{0:\infty} := \lim_{i\to\infty} \mathbb{S}_{0:i}$	the space of all RPs with a given root box

Statistical regular pavings

${}^{\circ}s$	a statistical regular paving (SRP)
$\mathbb{V}({}^{\circ}s)$	the set of all nodes of an SRP ${}^{\circ}s$
$\mathbb{L}({}^{\circ}s)$	the set of all leaf nodes of an SRP ${}^{\circ}s$
$\mathbb{L}^{+}({}^{\circ}s)$	the set of all non-empty nodes of an SRP ${}^{\circ}s$
$\mathbb{L}^{\triangledown}({}^{\circ}s)$	the set of all splittable leaf nodes of an SRP ${}^{\circ}s$
$\mathbb{C}({}^{\circ}s)$	the set of all cherry nodes of an SRP ${}^{\circ}s$
$\boldsymbol{x}_{\mathbb{L}({}^{\circ}s)}$	the partition of a root box $\boldsymbol{x}$ described by SRP ${}^{\circ}s$
${}^{\circ}\mathbb{S}_{i:j}$	the set of all SRPs with a given root box and k splits where $k \in \{i, i+1, \ldots, j\}$
${}^{\circ}\mathbb{S}_{0:\infty}$	the space of all SRPs with a given root box
${}^{\circ}\tilde{\mathbb{S}}$	a finite space of SRPs with a given root box
${}^{n}X$	a sample of size n associated with an SRP
${}^{\subset n}X_{\rho\mathsf{v}}$	the subset of ${}^{n}X$ in the box $\boldsymbol{x}_{\rho\mathsf{v}}$ of a node $\rho\mathsf{v}$ in an SRP
$\rho\mathsf{v}_{[\downarrow]}$	the set of associations between an SRP node $\rho\mathsf{v}$ and ${}^{\subset n}X_{\rho\mathsf{v}}$
$\#\boldsymbol{x}_{\rho\mathsf{v}}$	the number of sample points associated with node $\rho\mathsf{v}$ in a SRP (size of ${}^{\subset n}X_{\rho\mathsf{v}}$)

Mapped regular pavings

${}^{\blacksquare}f$	a mapped regular paving (MRP)
${}^{\blacksquare}\mathcal{F}$	a class of MRPs with a given root box
$\mathbb{V}({}^{\blacksquare}f)$	the set of all nodes of an MRP ${}^{\blacksquare}f$
$\mathbb{L}({}^{\blacksquare}f)$	the set of all leaf nodes of an MRP ${}^{\blacksquare}f$
$\mathbb{C}({}^{\blacksquare}f)$	the set of all cherry nodes of an MRP ${}^{\blacksquare}f$
$\boldsymbol{x}_{\mathbb{L}({}^{\blacksquare}f)}$	the partition of a root box $\boldsymbol{x}$ described by MRP ${}^{\blacksquare}f$

Real-mapped regular pavings

${}^{\square}f$	an real-mapped regular paving (RMRP)
${}^{\square}\mathcal{F}$	a class of RMRPs with a given root box
$\mathbb{V}({}^{\square}f)$	the set of all nodes of an RMRP ${}^{\square}f$
$\mathbb{L}({}^{\square}f)$	the set of all leaf nodes of an RMRP ${}^{\square}f$
$\mathbb{C}({}^{\square}f)$	the set of all cherry nodes of an RMRP ${}^{\square}f$
$\boldsymbol{x}_{\mathbb{L}({}^{\square}f)}$	the partition of a root box $\boldsymbol{x}$ described by RMRP ${}^{\square}f$

Chapter 1

Introduction

1.1 Motivation

The probability density function f of a random variable $X \in \mathbb{R}^d$ is a function modelling the local concentration of probability mass (Devroye and Lugosi 2001, chap. 1). Specifying f gives a "natural description" of the distribution of X (Silverman 1986, p. 1). Density estimation is the construction of an estimate $\hat{f}$ from observed data assumed to be a sample drawn from a distribution with density f (Silverman 1986, chap. 1) and is a fundamental tool in the analysis of data in many fields (Sheather 2004). There are two general approaches: parametric density estimation and nonparametric density estimation. This dissertation is concerned with nonparametric density estimation: estimation of a density f without assuming that f is a member of a known parametric family.

The histogram and the kernel density estimate (KDE) are the two most commonly used forms of nonparametric density estimate (Wasserman 2007, chap. 6). Other density estimators include orthogonal series estimators (including wavelet estimators) and nearest neighbour estimators (Silverman 1986, chap. 2). Adaptations and specialisations of these density estimation methods may be used for particular types of data, high-dimensional data, and very large data sets (Scott and Sain 2005; Lee and Gray 2009). However it is formed, the density estimate is some smoothed representation of the observed data (Whittle 1958; Wasserman 2003, chap. 20). The density estimation method determines how this smoothing is performed. Data-adaptive density estimation methods adapt the amount of smoothing to the local density of the data (Silverman 1986, chap. 2).

Density estimates may be used for data visualisation, inference, estimation of functionals of the density, prediction and classification, and simulation (Silverman 1986, chap. 2). There is no universally 'best' method for all applications and contexts: a KDE may (given a suitable choice of smoothing parameters) be expected to give the most accurate estimate for a given sample size but the computational cost of attaining this level of accuracy may be very high (Gray and Moore 2003a). Data-adaptive smoothing is increasingly necessary to control estimation errors as the number of dimensions increases (Scott 1992, chap. 7), but is also increasingly computationally expensive and difficult to achieve (Scott and Sain 2005).

The ultimate aim of the density estimation process is an important factor in the choice of density estimation method. For example, density estimates used for data-mining applications are often optimised for fast lookup of point or range queries (Müller et al. 2009). In the case of data-visualisation, 1 or 2-dimensional density estimates can be displayed easily but with higher dimensional data other graphical displays must be used (Scott 1992, chap. 1) or projections of the density depicted using a number of marginal or conditional density

estimates: histograms may convey the same visual information about the shape of the density as KDEs and be considerably quicker to produce. If a density estimate is used as the basis for further inference then a whole series of operations may be required, including marginal and conditional estimates and highest density regions.

Regular paving (RP)s (Harlow et al. 2012) are a group of very general data structures developed to facilitate arithmetical operations on the objects represented using these structures. It has been shown that RPs can be used as the basis for data-adaptive multivariate histograms and other nonparametric data-adaptive piecewise-constant function (PCF) density estimates (Sainudiin et al. 2013). Sainudiin et al. (2013) developed a Metropolis-Hastings Markov chain method with a stationary distribution approximating the posterior distribution of RP-histograms and estimated the expectation under this posterior distribution by exploiting the arithmetic properties of RPs to average samples from the chain. The RP structure places some restrictions on the density estimate, including the restriction on the form of the estimate to the class of PCFs, but has the following advantages:

- A wide range of operations can be carried out very efficiently on an RP-based PCF density estimate. These efficient operations include the creation of marginal and conditional density estimates, and highest density sets, directly from the density estimate.
- A collection of compatible RP-based histogram density estimates can be averaged. The result is a smoothed RP-based PCF density estimate with better consistency properties than the histograms but all the operational capabilities of an RP-based PCF.
- The RP structure is very general; RP-based PCF density estimates can potentially be created in many different ways in addition to those covered in Sainudiin et al. (2013).

Sainudiin et al. (2013) tested the averaged RP-histogram density estimate with uniform data in up to 1,000 dimensions and found that the method coped well with this type of high-dimensional unstructured data. Results using data simulated from uniform mixture approximations to multivariate Gaussian and Rosenbrock densities showed that the number of dimensions in which the method is computationally feasible with these examples of more structured data is much lower, about $d = 5$ or $d = 6$, that very large data sets may be needed, and that the estimated mean integrated absolute error (MIAE)s are considerably higher compared with estimates for unstructured data. Many conventional kernel density estimation methods are only effective with data in less than five or six dimensions (Gray and Moore 2003a; Zhang et al. 2006) but no comparison of the accuracy of RP-structured density estimates against KDEs was attempted in Sainudiin et al. (2013).

The averaged histogram RP-based PCF (Sainudiin et al. 2013) therefore has some attractions as a density estimation method in up to around five dimensions in situations where there is a large amount of sample data available and where it is advantageous to be able to carry out subsequent operations efficiently, directly on the density estimate itself. There is also

potential to try to combine the advantages of other density estimation methods, for example the desirable statistical properties of KDEs (including rate of convergence of the estimate to the true density) with the computational efficiency of operations on RPs by approximating other density estimates with RP-based structures.

Even in a simple data visualisation application the ability to create marginal and conditional density estimates quickly is useful. The motivating example in this dissertation is the more challenging task of simulation-intensive Bayesian inference for complex models with intractable likelihood functions. Simulations are used to support inference in many 'real-world' applications where the models can be complex, the numbers of parameters large, and only partial observations may be available or the observations may be strongly affected by process noise. In these situations many inference problems cannot be satisfactorily solved by analytical methods because the required likelihood functions cannot be expressed as tractable formulae, but it is relatively easy to exploit computing power to simulate large amounts of data from a very complex stochastic model (Hartig et al. 2011). RP-based density estimates may be particularly well-suited to these applications.

The methods most successfully used to tackle this type of inference problem obtain an estimate of a posterior density of the parameters of interest given a sample of observed data by simulating directly from an approximation to that target posterior. The arithmetical properties of an RP-structured density estimate can be exploited to give an alternative method for obtaining an estimate of the posterior density by creating an estimate of the joint density of the parameters and data, again from simulations, and obtaining the posterior as a conditional function of this joint density. Posterior density estimates for independent observations can be combined to give an estimate of the posterior given a number of independent samples of the data. The construction and analysis of a density estimate over the joint data-parameter space also facilitates richer insights into the relationships between the data and the parameters. The method is described in detail in Chapter 9 and the computational properties of RP-structured density estimates are demonstrated in the major examples contained in that chapter.

1.2 Objectives

The overall purpose of this thesis is to explore the potential for density estimation using RPs. The specific objectives of this thesis are to:

- Formalise the operational properties of RP-structured density estimates.
- Augment the Metropolis-Hastings Markov chain process developed in Sainudiin et al. (2013) with improved methods for assessing convergence.
- Carry out a more thorough investigation of the strengths and weaknesses of the averaged RP-histogram estimate, including a comparison with KDEs.

- Show how the an RP-structured density estimate can be used to approximate a KDE.
- Demonstrate the properties of RP-structured density estimates in the context of the motivating example, Bayesian inference for complex models with intractable likelihood functions.

1.3 Outline of thesis

Chapter 2 (Background and literature review) contains a summary of statistical concepts used in this thesis and a review of multivariate density estimation methods.

Chapter 3 (Regular pavings) defines the paving structures used in this thesis. The basic RP data structure and the generalisation of an RP to a mapped regular paving (MRP) are described. A form of RP called a statistical regular paving (SRP), capable of analysing data samples, is also defined and the process for creating a histogram density estimate using an SRP is outlined.

Chapter 4 (Data-adaptive partitioning using statistical regular pavings) gives an overview of data-adaptive density estimation using SRPs.

Chapter 5 (Randomised priority queues and statistical regular pavings) describes SRP partitioning using a randomised priority queue (RPQ).

Chapter 6 (Markov chain Monte Carlo partitioning and statistical regular pavings) discusses Markov chain Monte Carlo (MCMC) methods for creating an estimate of the posterior expectation of the distribution of SRP histograms.

Chapter 7 (Regular paving approximation of a kernel density estimate) shows how a KDE can be approximated using an RP-based structure and compares averaged SRP histogram density estimates with KDEs.

Chapter 8 (Simulation-intensive inference) reviews Bayesian methods for simulation-intensive parametric inference, the motivational application explored in Chapter 9.

Chapter 9 (RPABC: Regular pavings for simulation-intensive inference) describes how RP-structured density estimates can be applied to simulation-intensive Bayesian parametric inference problems. The chapter includes examples that demonstrate both the strengths and weaknesses of RPs-structured density estimates in this context.

Appendix A gives details of the computer code developed as part of this thesis together with the hardware and other software used to obtain the results and figures in this thesis.

Chapter 2

Background and literature review

2.1 Statistical inference

The following definitions are taken from Wasserman (2003, chap. 6):

Statistical model: A set $\mathfrak{F}$ of distributions (or densities, or regression functions).

Parametric model: A set $\mathfrak{F}$ that can be parameterised by a finite number of parameters.

Nonparametric model: A set $\mathfrak{F}$ that cannot be parameterised by a finite number of parameters.

Statistical inference: The process of using data to infer something about the distribution that generated the data.

Inference using as few assumptions as possible, or inference using a nonparametric (infinite-dimensional) model, is described as *nonparametric* inference (Wasserman 2007, chap. 1). Inference assuming some parametric model is described as *parametric inference*.

Sections 2.1.1 and 2.1.2 below review the fundamentals of inference from a classical (frequentist) and Bayesian perspective. The content of these sections establishes the notation to be used in the remainder of this thesis.

2.1.1 Likelihood

Let $\mathfrak{F}$ be a set or family of parametric models with parameter space Θ. Each $\boldsymbol{f} \in \mathfrak{F}$ is indexed by a unique $\theta \in \Theta$. If it is assumed that $X \sim \boldsymbol{f} \in \mathfrak{F}$ but the value of θ is not known then the Cartesian product of the sample space $\mathbb{X}$ and the parameter space Θ is the *inference universe* and $f(x, \theta)$ is the *sampling distribution surface*, a surface defined on the inference universe Bolstad (2010, chap. 1).

The notation developed in this section and used throughout this thesis distinguishes between symbols representing function arguments, such as *some* value $x \in \mathbb{X}$ that *may* be taken by a random variable X or some $\theta \in \Theta$ that *may* index a model $f \in \mathfrak{F}$, and known values such as a realisation $x^\star$ of X or a specific value $\theta^\star$ for θ used in a model. For example, $f(x; \theta^\star)$ is a function of x; $f(x^\star; \theta^\star)$ is a *value*, an evaluation of $f(x; \theta^\star)$ at $x^\star$; $f(x^\star \mid \theta)$ is a function of θ. The distinction may unnecessary in the context of the simple concepts discussed here but is useful to clarify the interactions of real observed data, simulations from a model, and operations on density estimates that are all involved in the simulation-intensive inference methods discussed in Chapters 8 and 9.

For a specific value $\theta^\star$ of θ, $f(x;\,\theta^\star)$ is a probability density (or mass) function for $x \in \mathbb{X}$. For a specific value $x^\star$ of X, $f(x^\star;\,\theta)$ is a function of θ for $\theta \in \Theta$ and is *not* a probability density (mass) function. The likelihood function $\mathcal{L}(\theta)$ is any function of θ that is proportional to $f(x^\star;\,\theta)$ (Sorensen and Gianola 2002, chap. 3)

If $X_1, \ldots, X_n$ have a joint distribution parameterised by θ with density $f(X_1 = x_1, \ldots, X_n = x_n;\,\theta)$, and the corresponding realisations $x_1^\star, \ldots, x_n^\star$ are observed, then the likelihood function $\mathcal{L}_n(\theta) : \Theta \to [0, \infty)$ is given by:

$$\mathcal{L}_n(\theta) \;\propto\; f(x_1^\star, \ldots, x_n^\star;\,\theta) \;\;.$$

If $X_1, \ldots, X_n \overset{\text{iid}}{\sim} f(X_1 = x_1;\,\theta)$ then the *product likelihood* is

$$\begin{aligned} \mathcal{L}_n(\theta) &\propto f(x_1^\star, \ldots, x_n^\star;\,\theta) \\ &= f(x_1^\star;\,\theta) \times \ldots \times f(x_n^\star;\,\theta) \\ &= \prod_{i=1}^{n} f(x_i^\star;\,\theta) \;\;. \end{aligned} \tag{2.1}$$

The log-likelihood is $\ell(\theta) = \log \mathcal{L}(\theta)$. A maximum likelihood estimate (MLE) $\widehat{\theta}_{MLE}$ for θ is a value of θ at which the likelihood function $\mathcal{L}(\theta)$ reaches its highest point.

2.1.2 Posterior distribution

From a Bayesian perspective, for inference about the same $\mathfrak{F}$ with parameter space Θ, the parameter θ is considered as a random variable with sample space Θ. The model conditional on θ has density function $f(x \,|\, \theta) \in \mathfrak{F}$. A prior distribution for θ with probability density function $f(\theta)$ can be combined with $f(x \,|\, \theta)$ to give a joint density function $f(x,\,\theta) : \mathbb{X} \times \Theta \to [0, \infty)$:

$$f(x,\,\theta) \;=\; f(x \,|\, \theta)\, f(\theta) \;\;.$$

Unlike the sampling distribution surface, the joint distribution is a probability distribution.

Given a realisation $x^\star$ of X, the conditional distribution $f(\theta \,|\, x^\star)$ is a function of θ for $\theta \in \Theta$ and again this is a probability density (mass) function. This conditional density is the density function of the posterior distribution of θ given $x^\star$. Formally,

$$f(\theta \,|\, x^\star) \;=\; \frac{f(x^\star \,|\, \theta) f(\theta)}{\int_\Theta f(x^\star \,|\, \theta) f(\theta)\, \mathrm{d}\theta} = \frac{f(x^\star \,|\, \theta) f(\theta)}{f(x^\star)} \;\;,$$

and $f(x^\star)$, or the evaluation of the marginal density $f(x)$ at $x = x^\star$, is a constant and so $f(\theta \,|\, x^\star) \propto f(x^\star \,|\, \theta)\, f(\theta)$.

If $X_1, \ldots, X_n \overset{\text{iid}}{\sim} f(x \,|\, \theta)$, and the corresponding realisations $x_1^\star, \ldots, x_n^\star$ are observed, then $f(x_1^\star, \ldots, x_n^\star \,|\, \theta) = \prod_{i=1}^n f(x_i^\star \,|\, \theta)$. The symbolic form $\mathcal{L}_n(\theta)$ is typically still used for functions

proportional to $\prod_{i=1}^{n} f\left(x_i^\star \mid \theta\right)$, so

$$\begin{aligned} f\left(\theta \mid x_1^\star, \ldots, x_n^\star\right) &= \frac{\mathcal{L}_n(\theta) f(\theta)}{\int_\Theta \mathcal{L}_n(\theta) f(\theta)\, \mathrm{d}\theta} \\ &\propto \mathcal{L}_n(\theta) f(\theta) \ . \end{aligned} \tag{2.2}$$

2.2 Density estimation

Given a random variable $X \in \mathbb{R}^d$ with unknown density function f, and an independent and identically distributed sample X_1, X_2, ... X_n drawn from f, a density estimate of f is a mapping $\hat{f}_n : \mathbb{R}^d \times (\mathbb{R}^d)^n \to \mathbb{R}^d$ (Devroye and Lugosi 2001, chap. 1). In general an estimate of a density f may also be denoted by $\hat{f}$ (Klemelä 2009, chap. 9), not explicitly showing the dependence of $\hat{f}$ on the sample of size n.

There are two general approaches to density estimation: parametric density estimation and nonparametric density estimation. This thesis is concerned with nonparametric density estimation (estimation of a density f without assuming that f is a member of a known parametric family). Nonparametric density estimators use at least one smoothing parameter (Scott 1992, chap. 4) to form the smoothed representation of the data that constitutes the density estimate (Whittle 1958; Wasserman 2003, chap. 20). An undersmoothed density estimate (too little smoothing) is too noisy, the variation of the individual observation obscuring underlying structure of the whole sample (Bowman and Azzalini 1997, chap. 1). An oversmoothed density estimate, however, under-represents (smooths out) important features of the data that reflect the true underlying density. The density estimation problem is to find an appropriate level of smoothing when the true density is unknown.

Typically, more smoothing (larger values of smoothing parameters) may be needed where the sample data is sparsest; less smoothing where it is most concentrated. Smaller values of the smoothing parameters are also required to be able to properly reflect sharper peaks and troughs in the underlying density (Sain 2002). *Locally adaptive* density estimators allow the smoothing parameters to vary according to the local features of the data. Multivariate density estimation may require the smoothing parameters to vary between coordinates (Klemelä 2009, chap. 14).

Histograms and kernel density estimate (KDE)s are the two most common forms of nonparametric density estimate for data assumed to be drawn from a continuous distribution. Both can be used for univariate and multivariate data. Histogram density estimation is reviewed in Section 2.4. Section 2.5 gives a brief overview of kernel density estimation methods.

2.3 Estimation error

The estimation error is some measure of the discrepancy or distance between the estimate $\hat{f}$ and the true density f. Four measures of the estimation error are commonly found:

L_1 **distance:** The L_1 error is $\int |f - \hat{f}|$. The L_1 error is also known as the integrated absolute error (IAE) (Scott 1992, chap. 2).

L_2 **distance:** The L_2 error is $\int \left(f - \hat{f}\right)^2$. The L_2 error is also known as the integrated squared error (ISE) (Scott 1992, chap. 2).

Hellinger distance: The Hellinger distance is $\int \left(f^{\frac{1}{2}} - \hat{f}^{\frac{1}{2}}\right)^2$ (Scott 1992, chap. 2).

Kullback-Leibler loss: The Kullback-Leibler loss is $\int \hat{f} \log \frac{\hat{f}}{f}$. The Kullback-Leibler loss is a measure of the expected likelihood ratio error and is closely related to information-theoretic entropy criteria (Pawitan 2001).

The L_1 error is invariant under any monotone transformation; this means that errors for estimates of different densities can be compared on an absolute scale. There is also a direct interpretation of the L_1 error in terms of the maximum possible difference in the probability of a set A under f and under $\hat{f}$ (Scott 1992; Devroye and Lugosi 2001).

The rate of convergence of a density estimator measures how quickly the density estimate converges to the true density as the sample size increases (Sheather 2004). A density estimate is said to be strongly L_1 consistent if the L_1 error tends to 0 asymptotically, i.e., if the same general form of rule is used to make the estimate for any sample size then as the sample size $n \to \infty$, the L_1 error $\to 0$. More formally, if the sequence of estimates for increasing sample size n is $\{\hat{f}_n\}$, $n = 1, 2, \ldots$, then the sequence is said to be strongly L_1 consistent if $\int |\hat{f}_n - f| \to 0$ with probability 1 as $n \to \infty$ (Lugosi and Nobel 1996).

2.4 Histograms

A histogram is based on a partition of the data space; the elements of the partition are commonly known as bins. The choice of bin width(s) is the smoothing parameter: wider bins give more smoothing, narrower bins less smoothing. The bins of a *regular* histogram are all equally-sized; the bins of an *irregular* histogram can vary in size.

Taking the general case of a (possibly irregular) histogram density estimate based on n observations $x_1^\star, \ldots, x_n^\star$ in $\mathbb{R}^d$, let the histogram partition consist of m bins $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m$. Let $\#\boldsymbol{x}_j$ be the number of the n observations falling into bin $\boldsymbol{x}_j$, and $\mathrm{vol}\,(\boldsymbol{x}_j)$ be the volume of bin j. For any $x \in \mathbb{R}^d$, let the indicator $\mathbb{1}_{\boldsymbol{x}_j}(x) = 1$ if $x \in \boldsymbol{x}_j$; 0 otherwise. Then the histogram

estimate $\hat{f}_n$ is given by:

$$\hat{f}_n(x) = \sum_{j=1}^{m} \mathbb{1}_{\boldsymbol{x}_j}(x) \left(\frac{\#\boldsymbol{x}_j}{n\text{vol}\left(\boldsymbol{x}_j\right)} \right) \quad . \tag{2.3}$$

The histogram likelihood is

$$\mathcal{L}\left(\hat{f}_n\right) = \prod_{j=1}^{m} \left(\frac{\#\boldsymbol{x}_j}{n\text{vol}\left(\boldsymbol{x}_j\right)} \right)^{\#\boldsymbol{x}_j} \quad . \tag{2.4}$$

Given a specified partition, the histogram is a maximum likelihood density estimator over the class of simple (piecewise-constant) functions with the same partition (Scott and Sain 2005). However, considered over different partitions, the histogram likelihood $\to \infty$ as bin volume $\to 0$.

A regular histogram density estimate can be shown to be strongly L_1 consistent (Devroye and Györfi 1992). Wand (1997) and Birgé and Rozenholc (2006) give important results for optimal partitions for regular univariate histogram using the L_2 error. Regular partitioning with small enough bins to suit the modes of the density will give too many bins in low or flat density areas (Rissanen et al. 1992). Regular partitioning with a bin width more suited to the overall variability of the data may compromise the potential of the histogram to show important local features in the highest density areas. Multivariate histograms with a single bin width are not able to adapt to spatially varying smoothing requirements (Klemelä 2009, chap. 17).

The choice of origin, or the point(s) determining bin edges can also be important (Silverman 1986). The average shifted histogram (ASH) density estimate (Scott 1992, chap. 5) attempts to deal with the influence of the choice origin but is not a data-adaptive estimator. The ASH is a (possibly weighted) average over a number of histograms with different bin origins. The range of bin origins to use, the bin widths, and the weighting method, must all be specified as inputs to the algorithm. The piecewise-constant function (PCF) ASH estimate has better consistency properties than a histogram estimate with the same complexity (number of elements in the PCF), but this complexity is entirely determined by inputs to the algorithm.

A data-dependent partition allows the bin width to vary in a way that is determined by the data. Stone (1985) showed that data-dependent partitions can provide estimates which are theoretically superior to those using partitions based simply on the number of data points in the data set. It can also be shown that, under certain conditions, a histogram density estimate with a data-dependent partition is strongly L_1 consistent. For a sequence of estimates with data dependent partitions $\{\hat{f}_n\}$, $\int |\hat{f}_n - f| \to 0$ with probability 1 as $n \to \infty$ (Lugosi and Nobel 1996) if all of the following conditions are met:

- the size of the partition (number of bins) grows sub-linearly relative to the size of the data set n;

- the partition grows sub-exponentially relative to n in terms of a combinatorial complexity measure; and
- the probability of any point in the data set being in a bin with largest diameter on any coordinate $> \gamma$ gets smaller for all $\gamma > 0$ as n increases (non-empty bins shrink).

There are a large number of different approaches to the problem of how to create data-dependent partitions. The common aim is to achieve some proper level of smoothing for each region of the data but to allow this to vary over the entire sample space. The differences come in what is meant by 'proper' (when to stop) and how each refinement of the partition is determined (how to continue).

A common approach is to use a form of penalised likelihood estimator such as the Akaike information criterion (AIC). Penalised likelihood methods can be used for selecting the optimal bin width for regular histograms (Taylor 1987; Birgé and Rozenholc 2006) and irregular multivariate histograms (Castellan 1999; Rozenholc et al. 2009). Both the AIC and the Bayesian information criteria (BIC) may be unsuitable criteria for histogram density estimation (Massert 2007; Rozenholc et al. 2009). Birgé and Rozenholc (2006) notes that in general optimality criteria that rely on an asymptotic estimate of risk (such as cross-validation) or any optimality criteria that depends on asymptotic performance (including many of the penalised likelihood approaches) do not necessarily perform well with small sample sizes. These penalty functions may also depend on assumptions about the unknown underlying density (Birgé and Rozenholc 2006). Rozenholc et al. (2009), Castellan (1999), and others use the Hellinger distance as a measure of risk which does not depend on asymptotic considerations.

Combinatorial minimum distance estimation methods for L_1 optimal bin-width selection are discussed in Devroye and Lugosi (2004) and Biau et al. (2007).

A statistically equivalent block (SEB) partition of a sample space is some partitioning scheme that results in equal numbers of data points in each element (block) of the partition (Tukey 1947) (except possibly in blocks on the boundary of the partitioned space). Loftsgaarden and Quesenberry (1965) seem to have provided the first treatment of consistent density estimates using statistically equivalent blocks, providing a consistent method for obtaining density estimates at individual points in a sample of multivariate data. Gessaman (1970) generalised this to a density estimate over an entire multidimensional space. Both Loftsgaarden and Quesenberry (1965) and Gessaman (1970) considered blocks created by 'cuts' orthogonal to a coordinate axis. Devroye et al. (1996, chap. 21) revisited this in the context of classification rules and discussed the conditions under which a classifier based on Gessaman's data-dependent partition is strongly consistent.

Knuth (2006b) describes a Bayesian framework and an algorithm for finding the equal-bin-width-based PCF estimate to maximise the marginal posterior probability of the number of bins by a brute-force search. This method is only practical when applied to small amounts of low-dimensional data.

Tree-based histograms are used in fields such image processing or data-mining to speed up storage and retrieval of data. A tree structure can also be used in algorithms for creating data-adaptive histograms. This is especially suitable where the algorithm uses some form of recursive partitioning strategy, again often in association with a penalty function to control complexity. A *greedy* partitioning algorithm makes locally optimal decisions (with respect to the chosen optimality criterion) based on the immediately available information in each step but is not guaranteed to find a globally optimal solution (Klemelä 2009, chap. 17). Several greedy data-adaptive tree-structured histogram algorithms have been developed, including methods that grow the tree (partition) step-by-step or that grow the tree to represent the most complex allowable partition and then use a greedy algorithm to prune to reduce the tree (reduce the number of elements in the partition (Klemelä 2009, chap. 17). The tree-structured HiRed histogram of Baltrunas et al. (2006) uses a similar recursive bisect-and-prune approach with an L_1 distance measure and a minimum bin volume to control complexity. Partitioning trees can also be used in non-greedy complexity-penalised optimisation algorithms that perform an exhaustive comparison of a limited set of possible partitions (Klemelä 2009, chap. 18).

One feature that all the above methods have in common is that they produce a point estimate of the target density. Bootstrapped aggregation methods create the density estimate as the average histogram over a number of bootstrapped samples from the original sample data. Bootstrapped aggregation of data-adaptive histograms can be used both to decrease the variance of the point estimate and, like ash density estimators, to obtain a more complex estimate with better consistency properties (Klemelä 2009, chap. 17).

In general, the methods described above are not well-suited to very high-dimensional data because the computational complexity of most density estimation algorithms grows exponentially with the dimensions (Scott 1992, chap. 7). Another important issue for density estimation for higher-dimensional data is that as the dimensions increase, relatively more of the sample will fall in the lower density regions (Silverman 1986, chap. 6). Effectively the total area of the tails (typically on the edges of the domain) represents an increasing proportion of the total space. This gives rise to the seemingly-paradoxical 'empty-space' phenomenon (Scott 1992, chap. 7) that regions of theoretically high density can contain relatively few of the observations in the sample. Regular histograms do not cope well with this (Scott 1992, chap. 3) and data-adaptive histograms with bin-widths that can adapt to the local density of the data are especially important in higher dimensions.

Some of the structures developed for data-mining applications such as the multi-resolution *kd*-trees and the ball trees of Gray and Moore (2003b) cope better in high dimensions, especially if the underlying density is highly structured.

Many variations on the histogram as ways to summarise and organise very large data sets have been created for data-mining, image management, and database management. Often

the efficiency of the structure is at least as important as its theoretical estimation error and consistency properties (see, for example, Rübel et al. (2008)). Fast look-up for queries, including conditional queries, is typically extremely important (Rübel et al. 2008) but other criteria may apply for, for example, histograms as the basis for hierarchical clustering structures for pattern discovery and data visualisation (Jović et al. 2004) and histograms for summarising fast-arriving streams of data (Guha and Koudas 2002).

2.5 Kernel density estimates

The general multivariate kernel estimator for n sample points $x_1, \ldots, x_n \in \mathbb{R}^d$ data is given by:

$$\hat{f}_K = \frac{1}{n|H|} \sum_{i=1}^{n} K(H^{-1}(x - x_i)) \ ,$$

where $K : \mathbb{R}^d \to \mathbb{R}$ is a kernel satisfying certain conditions (Scott (see 1992, chap. 6)) and H is a $d \times d$ nonsingular matrix. The product kernel is

$$\hat{f}_K = \frac{1}{nh_1 \ldots h_d} \sum_{i=1}^{n} \left(\prod_{j=1}^{d} K \left(\frac{x_j - x_{ij}}{h_j} \right) \right) \ ,$$

where $h_1, \ldots, h_d$ are the kernel estimation bandwidths (smoothing parameters), one for each dimension, and the same univariate kernel K is used in each dimension.

Locally adaptive kernel density estimators (Scott and Sain 2005) include some form of variable (data-dependent) bandwidth allowing more smoothing in the tails of a distribution and less in high density areas.

In theory, a kernel density estimate has better consistency properties than a histogram density estimate (Wasserman 2007, chap. 6): the estimation error decreases faster as the sample size n increases (or, for a given sample size, a KDE can be expected to give a more accurate estimate of the true density than a histogram). However, the choice of bandwidth(s) is crucial (Scott and Sain 2005). If the domain of the unknown density function is bounded the consistency of the KDE is affected. Several different techniques may be used to try to deal with this (Karunamuni and Alberts 2005).

The 'Normal reference rule' (Scott 1992; Zhang et al. 2006) for the bandwidth for a Normal product kernel estimate originally developed for univariate data has a multivariate equivalent

$$h_j = \sigma_j \left(\frac{4}{n(d+2)} \right)^{\frac{1}{(d+4)}}$$

where σ_i is the standard deviation of coordinate j of the data, $j = 1, \ldots, d$. This will tend to give oversmoothed estimates for data where the true distribution is not reasonably similar to a

multivariate Normal and, although it is often used, can perform very poorly with non-Normal data (Scott 1992; Zhang et al. 2006).

The amount of research on optimal bandwidths for multivariate data has increased considerably over the past decade. Devroye and Lugosi (2001) describe the L_1 view, but the most common criterion for the estimation error is an L_2 measure: the ISE or mean integrated squared error (MISE). Methods for calculating plug-in estimates for the bandwidth matrix (for example, Duong and Hazelton (2003), Wand and Jones (1994)) or choosing optimal bandwidths (often using a cross-validated maximum likelihood criterion) have been developed (for example, Duong and Hazelton (2003); Scott and Sain (2005)), but most of the investigation and practical work has still tended to focus on bivariate densities (Duong and Hazelton 2005). Zhang et al. (2006) showed that a diagonal bandwidth matrix chosen using a Bayesian Markov chain Monte Carlo method and likelihood cross-validation can perform relatively well (compared with other commonly-used methods for choosing the bandwidth matrix) in at least five dimensions. Filippone and Sanguinetti (2011) used an approximate Bayesian approach to get similar estimation performance with less computation time.

With higher dimensional data there is an increasing need for locally adaptive bandwidths. As with multivariate histograms, this is due to the increasing importance of the tails of the distribution and the associated 'empty-space' phenomenon already discussed in Section 2.2. More smoothing (higher bandwidth) is needed in the low-density regions but if higher bandwidths are applied globally then features near the modes can be smoothed into invisibility (Sain 2002). For univariate data or a very low number of dimensions it is not uncommon to find that a KDE with a fixed bandwidth matrix is undersmoothed in the tails. Most of the sample points are near the mode(s) and the bandwidth(s) chosen using the overall variability of the sample will be lower than is suitable for the sparser data in the tails. In higher dimensions a fixed bandwidth matrix based on the overall variability of the data can give the opposite effect, with large bandwidths and serious oversmoothing around the mode(s), but it has proven to be very difficult to establish how to vary the bandwidths of multivariate kernel density estimator to take local variability into account, and in particular how to do this in practice with a computationally feasible estimation method (Sain 2002). Scott (1992, p. 202) suggested that in theory "kernel density estimation beyond 5 dimensions is fruitless", but also noted that the empirical results often belie this pessimistic statement.

Many of the KDE methods in general use do not scale well for large data sets and higher dimensions (Gray and Moore 2003b; Duong and Hazelton 2005). Binning (gridding) the data and the fast-Fourier transform can be used to speed up the computation time considerably (Wand 1994; Fan and Marron 1994) but binning may introduce additional approximation errors (Raykar et al. 2010) and these methods still scale badly and are usually restricted to low dimensional data (Gray and Moore 2003a). Cheng et al. (2006) describe a fast multivariate kernel density estimator based on binned data (Fan and Marron 1994) suitable for data sets

of at least a million data points but gave no timing or estimated error results for tests with bivariate Normal data described in the paper. The hierarchically-structured multi-bandwidth ball-trees of Gray and Moore (2003b) overcome some of the computational problems posed by large data sets at the expense of considerable computer memory consumption, and with some performance and error bound issues (Lee and Gray 2009). Lee and Gray (2009) combined a hierarchical structure with fast Gauss transforms (fast multipole-type methods) to give a very efficient method suitable for low to moderate dimensions; Lee et al. (2006) extended this to much higher dimensions using a new hierarchical structure and Monte Carlo estimates of error bounds. Multipole-inspired techniques have also been used in other fast KDE methods (Raykar et al. 2010).

2.6 Marginal and conditional density estimates

Marginal density estimates are usually made by applying the chosen density estimator to the sample data on the subset of coordinates of interest (the marginal coordinates). If multiple marginal density estimates are required, on different subsets of coordinates, the density estimation process is repeated multiple times. Computation of marginal histogram density estimates directly from a histogram estimate of a joint density is also straightforward if regular bin widths are used on each coordinate (the marginal histogram will have exactly the same bin widths on the marginal coordinates as does the joint histogram, and the marginal counts can found by summing counts over the coordinates marginalised out).

Conditional density estimation is more challenging. Rosenblatt's conditional density estimate for a random variable Y conditional on the value of a univariate random variable $X = x$ using KDEs is $\hat{f}\,(y \,|\, x) = \frac{\hat{f}_K(x,y)}{\hat{f}_K(x)}$ where $\hat{f}_K(x, y)$ is the KDE of $f\,(x,\, y)$ and $\hat{f}_K(x)$ is the KDE of the marginal density $f(x)$ (Bashtannyk and Hyndman 2001). Modifications to this estimate to correct for bias and select the KDE bandwidths using bootstrapped estimates are discussed in Bashtannyk and Hyndman (2001). Cross-validation methods for bandwidth selection have also been developed (see, for example, Hall et al. (2004) and Fan and Yim (2004)). Cheng et al. (2006) proposed a method for producing fast conditional and marginal functions by exploiting the data summaries used in their binned-data KDE for massive data sets. Györfi and Kohler (2007) considered the L_1 consistency properties of conditional histogram estimates from a theoretical point of view.

Chapter 3

Regular pavings

3.1 Introduction

A regular paving (RP) (Samet 1990; Jaulin et al. 2001; Kieffer et al. 2001) is a finite succession of bisections that partition a box $\boldsymbol{x}$ in $\mathbb{R}^d$ into sub-boxes using a tree-based data structure. Section 3.2 summarises the principle reasons for using a tree-based structure; Section 3.3 formally defines the RP structure.

A statistical regular paving (SRP) (Section 3.4) is an RP structure able to act as an partitioned 'container' and summariser for multivariate data.

A mapped regular paving (MRP) (Section 3.5) is an extension of an RP designed to facilitate arithmetical operations on the data structure itself. An $\mathbb{R}$-MRP or real-mapped regular paving (RMRP) (Section 3.6) can be used to represent a PCF. An SRP can be used to create a histogram density estimate represented as an RMRP.

3.2 Tree structures

Partitions of multi-dimensional space are usually represented using hierarchical data structures such as trees (Samet 2006). The main advantages are:

- Operations on the data structures are often well suited to spatial divide-and-conquer methods and hence to hierarchical data structures;
- A tree provides $\mathcal{O}(\log m)$ access time to any sub-box in a collection of m sub-boxes, regardless of the number of dimensions;
- Trees provide low-cost (constant time) insertion and deletion of sub-boxes, without the need to reallocate existing partitions in memory;
- Algorithms operating on trees can be expressed naturally and succinctly in a recursive form, with a correspondingly simple programming implementation (Kruse 1987).

3.3 Regular Pavings

3.3.1 Definitions

This section starts with some preliminary definitions of intervals, boxes, and a regular bisection of a box.

Let $\boldsymbol{x} := [\underline{x}, \overline{x}]$ be a compact real interval with lower bound $\underline{x}$ and upper bound $\overline{x}$ where $\underline{x} \leq \overline{x}$. Let the space of such intervals be $\mathbb{IR}$. The width of an interval $\boldsymbol{x}$ is $\mathrm{wid}\,(\boldsymbol{x}) := \overline{x} - \underline{x}$. The midpoint is $\mathrm{mid}\,(\boldsymbol{x}) := \frac{\underline{x}+\overline{x}}{2}$. A box of dimension d with coordinates in $\Delta := \{1, 2, \ldots, d\}$ is an interval vector:

$$\boldsymbol{x} := [\underline{x}_1, \overline{x}_1] \times \ldots \times [\underline{x}_d, \overline{x}_d] =: \boxtimes_{j \in \Delta} [\underline{x}_j, \overline{x}_j] \ .$$

The set of all such boxes is $\mathbb{IR}^d$, i.e., the set of all interval real vectors in dimension d. Consider a box $\boldsymbol{x}$ in $\mathbb{IR}^d$. Define the index ι to be the first coordinate of maximum width:

$$\iota := \min \left(\underset{i}{\mathrm{argmax}} (\mathrm{wid}\,(\boldsymbol{x}_i)) \right) \ .$$

A *bisection* or *split* of $\boldsymbol{x}$ perpendicularly at the mid-point along this first widest coordinate ι gives the left and right child boxes of $\boldsymbol{x}$

$$\boldsymbol{x}_{\mathsf{L}} := [\underline{x}_1, \overline{x}_1] \times \ldots \times [\underline{x}_\iota, \mathrm{mid}\,(\boldsymbol{x}_\iota)) \times [\underline{x}_{\iota+1}, \overline{x}_{\iota+1}] \times \ldots \times [\underline{x}_d, \overline{x}_d] \ ,$$

$$\boldsymbol{x}_{\mathsf{R}} := [\underline{x}_1, \overline{x}_1] \times \ldots \times [\mathrm{mid}\,(\boldsymbol{x}_\iota), \overline{x}_\iota] \times [\underline{x}_{\iota+1}, \overline{x}_{\iota+1}] \times \ldots \times [\underline{x}_d, \overline{x}_d] \ .$$

Such a bisection is said to be *regular*.

Note that this bisection gives the left child box a half-open interval $[\underline{x}_\iota, \mathrm{mid}\,(\boldsymbol{x}_\iota))$ on coordinate ι so that the intersection of the left and right child boxes is empty.

This refinement is a necessary condition for some of the operations described in Sections 3.5 and 3.6 and causes no complications for those operations that do not require it. If $\boldsymbol{x}$ is not a closed box in $\mathbb{IR}^d$ then it can be made into a closed box, if necessary, by taking its interval or box hull $\square\boldsymbol{x}$.

A recursive sequence of selective regular bisections of boxes, with possibly open boundaries, along the first widest coordinate, starting from the root box $\boldsymbol{x}$ in $\mathbb{IR}^d$ is known as a *regular paving* (Jaulin et al. 2001; Kieffer et al. 2001) or n-tree (Samet 1990) of $\boldsymbol{x}$.

A regular paving (RP) of $\boldsymbol{x}$ can also be seen as a binary tree formed by recursively bisecting the box $\boldsymbol{x}$ at the root node. Each node in the binary tree has either no children or two children. These trees are known as *plane binary trees* in enumerative combinatorics (Stanley 1999, Ex. 6.19(d), p. 220) and as *finite, rooted binary trees (frb-trees)* in geometric group theory (Meier 2008, Chap. 10). When the root box $\boldsymbol{x}$ is clear from the context we refer to an RP of $\boldsymbol{x}$ as merely an RP. Each node of an RP is associated with a sub-box of the root box that can be attained by a sequence of selective regular bisections.

Each node in an RP can be distinctly labelled by the sequence of child node selections from the root node. In the explanations, algorithms and diagrams that follow, the nodes may be labelled with strings composed of L (for 'left') and R (for 'right'). For example, a node labelled L is the left child of its parent.

The relationship of trees, labels and partitions is illustrated in Figure 3.1 using a simple one-dimensional example. The root node associated with root interval $\boldsymbol{x}_\rho \in \mathbb{IR}$ is labelled ρ. First, ρ is split into two child nodes. The split operation is denoted $\bigtriangledown_{\text{RP}}(\rho) = \{\rho\mathsf{L}, \rho\mathsf{R}\}$ and the left child and right child nodes are labelled $\rho\mathsf{L}$ and $\rho\mathsf{R}$, respectively. The left half of $\boldsymbol{x}_\rho$ that is now associated with node $\rho\mathsf{L}$ is labelled $\boldsymbol{x}_{\rho\mathsf{L}}$. Similarly, the right half of $\boldsymbol{x}_\rho$ that is associated with the right child node $\rho\mathsf{R}$ is labelled $\boldsymbol{x}_{\rho\mathsf{R}}$. $\rho\mathsf{L}$ and $\rho\mathsf{R}$ are a pair of *sibling nodes* since they share the same parent node ρ. A node with no child nodes is called a *leaf node*. A *cherry node* is a sub-terminal node with a pair of child nodes that are both leaves. This pair of sibling nodes can be *reunited* or *merged* to its parent cherry node ρ, thereby turning the cherry node into a leaf node. The merging operation is denoted $\bigtriangleup_{\text{RP}}(\rho\mathsf{L}, \rho\mathsf{R}) = \rho$.

Returning to Figure 3.1, the left node $\rho\mathsf{L}$ is split to get its left and right child nodes $\rho\mathsf{LL}$ and $\rho\mathsf{LR}$ with associated sub-intervals $\boldsymbol{x}_{\rho\mathsf{LL}}$ and $\boldsymbol{x}_{\rho\mathsf{LR}}$ respectively, formed by the bisection of interval $\boldsymbol{x}_{\rho\mathsf{L}}$ (because the root interval $\boldsymbol{x}_\rho$ is one-dimensional, each bisection is always on that single coordinate).

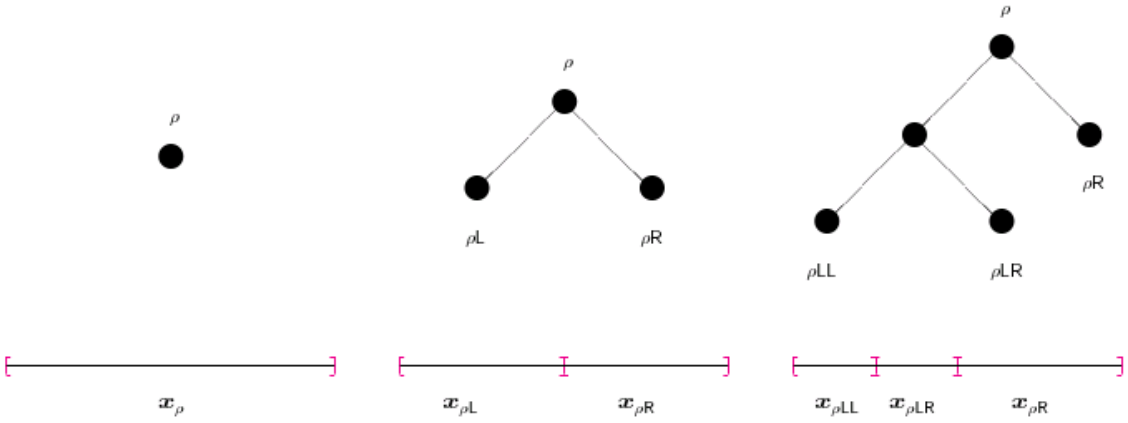


Figure 3.1: A sequence of selective bisections, starting from the root, produces an RP.

Figure 3.2 shows a sequence of bisections of a square (two-dimensional) root box. The first three bisections are the same as in Figure 3.1 (the *trees* are therefore identical), but Figure 3.2 shows the effect of always bisecting on the first widest coordinate. The first bisection, forming sub-boxes $\boldsymbol{x}_{\rho\mathsf{L}}$ and $\boldsymbol{x}_{\rho\mathsf{R}}$, takes place on the first widest coordinate of $\boldsymbol{x}_\rho$. This is the first coordinate because the box is square. The next bisection, of box $\boldsymbol{x}_{\rho\mathsf{L}}$ to form $\boldsymbol{x}_{\rho\mathsf{LL}}$ and $\boldsymbol{x}_{\rho\mathsf{LR}}$, takes place on the second coordinate because this is the first widest coordinate of $\boldsymbol{x}_{\rho\mathsf{L}}$.

The sequence is then extended with two further bisections. First the right child node $\rho\mathsf{R}$ is split into its child nodes $\rho\mathsf{RL}$ and $\rho\mathsf{RR}$ (again bisecting on the second coordinate of $\boldsymbol{x}_{\rho\mathsf{R}}$). Then $\rho\mathsf{LR}$ is selected for a final split, giving its child nodes $\rho\mathsf{LRL}$ and $\rho\mathsf{LRR}$. $\boldsymbol{x}_{\rho\mathsf{LR}}$ is square and is bisected on its first coordinate to form the sub-boxes $\boldsymbol{x}_{\rho\mathsf{LRL}}$ and $\boldsymbol{x}_{\rho\mathsf{LRR}}$.

Figures 3.1 and 3.2 also illustrate an important point about these RPs. Because of the restricted bisection rule (bisecting a box only at the mid-point along its first widest coordinate),

an RP tree will uniquely describe or encode the partition of some root-box $\boldsymbol{x}_\rho$ into sub-boxes. If there are two RPs with the same root box, and two nodes at exactly the same positions in their respective trees, then both of these nodes will have exactly the same box and can be considered to be 'equivalent'.

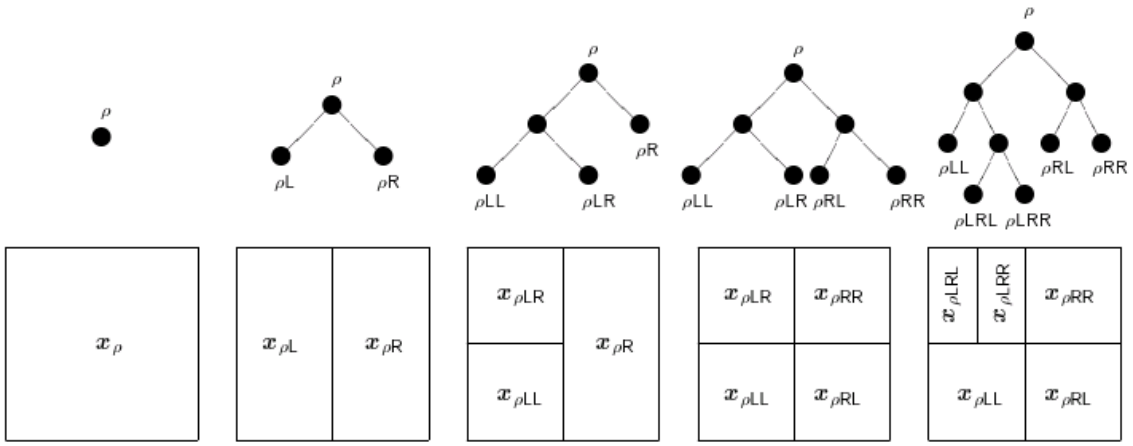


Figure 3.2: An extended sequence of selective bisections, 2-d root box.

Let the j-th interval of a box $\boldsymbol{x}_{\rho\mathsf{v}}$ be $[\underline{x}_{\rho\mathsf{v},j}, \overline{x}_{\rho\mathsf{v},j}]$. The volume of a d-dimensional box $\boldsymbol{x}_{\rho\mathsf{v}}$ associated with the node $\rho\mathsf{v}$ of an RP of $\boldsymbol{x}_\rho$ is the product of the side-lengths of the box.

$$\mathrm{vol}\left(\boldsymbol{x}_{\rho\mathsf{v}}\right) = \prod_{j=1}^{d} (\overline{x}_{\rho\mathsf{v},j} - \underline{x}_{\rho\mathsf{v},j}) \ .$$

The volume is associated with the depth of a node. The depth of a node $\rho\mathsf{v}$ in an RP is denoted by $\mathsf{d}_{\rho\mathsf{v}}$. A node has depth $\mathsf{d}_{\rho\mathsf{v}} = k$ in the tree if it can be reached by k splits from the root node. If an RP has root box $\boldsymbol{x}_\rho$ and a node $\rho\mathsf{v}$ in the RP has depth k, then the volume of the box $\boldsymbol{x}_{\rho\mathsf{v}}$ associated with that node is $\mathrm{vol}\left(\boldsymbol{x}_{\rho\mathsf{v}}\right) = 2^{-k}\mathrm{vol}\left(\boldsymbol{x}_\rho\right)$. This is because any split always results in the child node's box having half the volume of the parent node's box.

Any tree can be uniquely identified by the sequence of its leaf node depths if a consistent ordering of leaf nodes is used. For example the leaf nodes of the final RP in Figure 3.2, listed in left-to-right order, are $[\rho\mathsf{LL}, \rho\mathsf{LRL}, \rho\mathsf{LRR}, \rho\mathsf{RL}, \rho\mathsf{RR}]$. Where such labelling is used in this thesis, the leaf nodes are ordered left-to-right. Node $\rho\mathsf{LL}$ has depth 2, node $\rho\mathsf{LRL}$ has depth 3, etc. The sequence $2, 3, 3, 2, 2$ uniquely identifies the tree and the tree (as discussed above) uniquely identifies the partition of the root box, and so the sequence of leaf node depths also uniquely describes the partitioning of the root box $\boldsymbol{x}_\rho$.

In general, an RP is denoted by s. Where it is convenient, an RP may be labelled by its leaf node depth sequence. For example, the final RP in Figure 3.2 can be referred to as $s_{2,3,3,2,2}$.

The set of all nodes of an RP is denoted by $\mathbb{V} := \rho \cup \{\rho\{\mathsf{L}, \mathsf{R}\}^j : j \in \mathbb{N}\}$. The set of all leaf

nodes of an RP is denoted by $\mathbb{L}$. For the final RP in the sequence represented in Figure 3.2,

$$Lz(s_{2,3,3,2,2}) = \{\rho\mathsf{LL}, \rho\mathsf{RL}, \rho\mathsf{RR}, \rho\mathsf{LRL}, \rho\mathsf{LRR}\}$$

and

$$\mathbb{V}(s_{2,3,3,2,2}) = \{\rho, \rho\mathsf{L}, \rho\mathsf{R}, \rho\mathsf{LL}, \rho\mathsf{LR}, \rho\mathsf{RL}, \rho\mathsf{RR}, \rho\mathsf{LRL}, \rho\mathsf{LRR}\} \ .$$

The boxes associated with the leaf nodes of an RP are the partition of the root box $\boldsymbol{x}_\rho$. The set of leaf boxes of a regular paving s with root box $\boldsymbol{x}_\rho$ is denoted by $\boldsymbol{x}_{\mathbb{L}(s)}$. Let $\mathbb{S}_k$ be the set of all RPs with root box $\boldsymbol{x}_\rho$ made of k splits. Note that the number of leaf nodes $m = |\mathbb{L}(s)| = k+1$ if $s \in \mathbb{S}_k$.

The number of distinct binary trees with k splits is equal to the Catalan number C_k (Knuth 2006a).

$$C_k = \frac{1}{k+1}\binom{2k}{k} = \frac{(2k)!}{(k+1)!(k!)} \ . \tag{3.1}$$

For $i, j \in \mathbb{Z}_+$, where $\mathbb{Z}_+ := \{0, 1, 2, \ldots\}$ and $i \leq j$, let $\mathbb{S}_{i:j}$ be the set of RPs with k splits where $k \in \{i, i+1, \ldots, j\}$. The space of all RPs is then $\mathbb{S}_{0:\infty} := \lim_{j\to\infty} \mathbb{S}_{0:j}$. The size of the space of all RPs with between i and j splits, $|\mathbb{S}_{i:j}|$, is given by the sum of Catalan numbers:

$$|\mathbb{S}_{i:j}| = \sum_{k=i}^{j} C_k \ . \tag{3.2}$$

The size of the space of all RPs with up to k splits is $|\mathbb{S}_{0:k}|$.

Figure 3.3 displays the transition diagram over $\mathbb{S}_{0:3}$ where the gray arrows represent the transition from one RP state to another through a split or reunion. There may be more than one path from the root node to a particular RP in $\mathbb{S}_k$, i.e., more than one distinct sequence of k splits may result in the same RP in $\mathbb{S}_k$. In Figure 3.3, for example, there are two paths to $s_{2,2,2,2}$.

A computerised implementation of an RP will impose limits on the number of recursive bisections of a box in a regular partition. A node $\rho\mathsf{v}$ in an RP is only *splittable* if the two halves of the $\boldsymbol{x}_{\rho\mathsf{v}}$ that would result from a bisection are both properly computer representable. These limits are discussed in detail in Appendix C. The set of splittable nodes in an RP s is denoted by $\mathbb{L}^\triangledown(s)$.

3.3.2 Operations on regular pavings

The union of two RPs $s^{(1)}$ and $s^{(2)}$ in $\mathbb{S}_{0:\infty}$ with the same root box $\boldsymbol{x}_\rho$ is denoted by $s^{(1)} \cup s^{(2)}$. Intuitively, the leaf boxes of the union of two RPs can be seen as being obtained from overlaying or superimposing the partitions of the operand RPs as shown in Figure 3.4.

Let $\rho\mathsf{v}$ be a node of an RP s and let the Boolean function `IsLeaf`$(\rho\mathsf{v})$ return true if $\rho\mathsf{v}$ is a leaf node and false otherwise. Let `Copy`$(\rho\mathsf{v})$ return a copy of the RP tree rooted at node

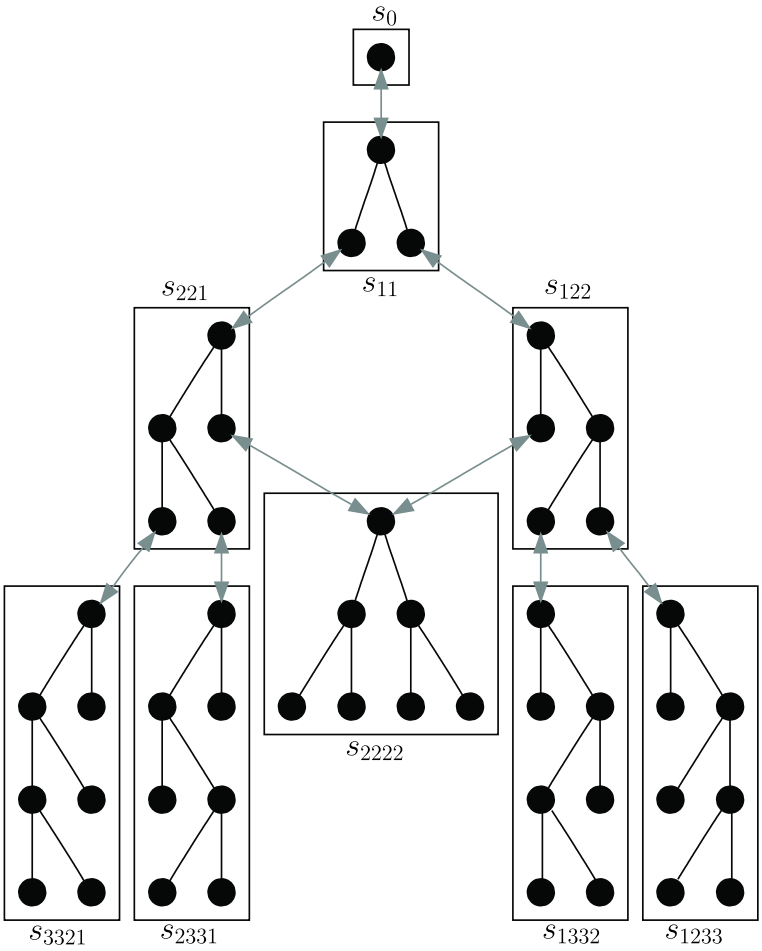
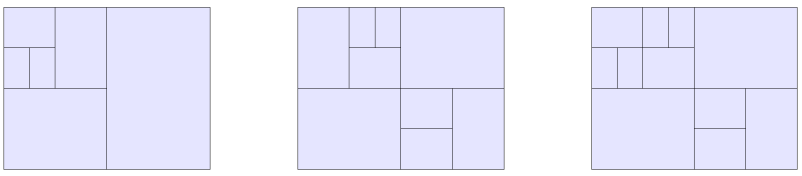


Figure 3.3: Transition diagram over $\mathbb{S}_{0:3}$ with split/reunion transitions from one RP state to another.



(a) $s^{(1)}$. (b) $s^{(2)}$. (c) $s^{(1)} \cup s^{(2)}$.

Figure 3.4: Union of two regular pavings of a root box in $\mathbb{R}^2$.

$\rho\mathsf{v}$. If $\rho\mathsf{v}$ is not a leaf node then let $\rho\mathsf{vL}$ and $\rho\mathsf{vR}$ be the left and right child nodes of $\rho\mathsf{v}$. Consider two RPs $s^{(1)}$ and $s^{(2)}$ with root nodes $\rho^{(1)}$ and $\rho^{(2)}$, respectively, and the same root box $\boldsymbol{x}_\rho$. Then, `RPUnion`$(\rho^{(1)}, \rho^{(2)})$ (Algorithm 3.1) returns the union of the two RP trees. The algorithm exploits the tree structure to recurse on pairs of nodes, moving through both trees simultaneously. Figure 3.5 shows two RPs with the same box $\boldsymbol{x}_\rho \in \mathbb{IR}^2$ and their union.

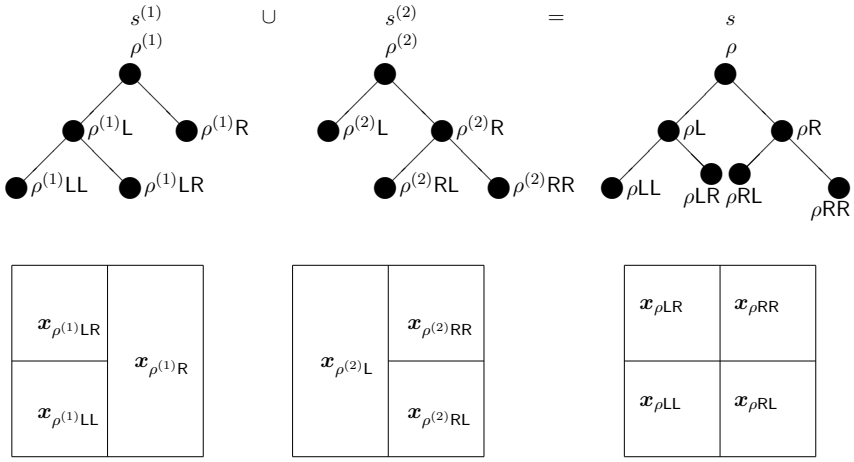


Figure 3.5: Union on the RPs $s^{(1)}$ and $s^{(2)}$.

The union operation $\cup$ is subject only to the restriction that the two operand RPs have the same root box. Remarkably, RPs are closed under such unions, i.e., if $s^{(1)}, s^{(2)} \in \mathbb{S}_{0:\infty}$ then $s^{(1)} \cup s^{(2)} =: s \in \mathbb{S}_{0:\infty}$.

The generality of the union operation is due to the restrictive bisection. Bisecting only on the first coordinate of maximum width means that, if some equivalent node $\rho\mathsf{v}*$ exists in both $s^{(1)}$ and $s^{(2)}$ (for example, $\rho\mathsf{v}^{(1)}\mathsf{L}$ and $\rho\mathsf{v}^{(2)}\mathsf{L}$ in Figure 3.5), then the boxes associated with these nodes ($\boldsymbol{x}_{\rho\mathsf{v}^{(1)}\mathsf{L}}$ and $\boldsymbol{x}_{\rho\mathsf{v}^{(2)}\mathsf{L}}$ in Figure 3.5) will be identical. Similarly, if some part of, say, $s^{(1)}$ is 'more partitioned' than that part of $s^{(2)}$ (as the left sub-box $\boldsymbol{x}_{\rho^{(1)}\mathsf{L}}$ of $s^{(1)}$ is in Figure 3.5), then that same partition can be exactly replicated in $s^{(2)}$ or a copy of $s^{(2)}$ by replicating in $\boldsymbol{x}_{\rho^{(2)}\mathsf{L}}$ the regular bisections encoded in the tree rooted at $\rho^{(1)}\mathsf{L}$.

3.4 Statistical regular pavings

3.4.1 Definition

A statistical regular paving (SRP) is an extension of the RP structure able to act as a partitioned 'container' and summariser for multivariate data. An SRP can be used to create a histogram of a data set. An SRP is effectively an association of a collection of data (the *data sample* or *data set*) with an RP-based structure where the nodes have additional properties:

- A node of an SRP tree can be associated with a subset of the sample data;

- A node of an SRP tree records recursively computable statistics relating to this sample subset.

An SRP is denoted by ${}^{\circ}s$. ${}^{\circ}\mathbb{S}_{i:j}$ is the set of all statistical regular pavings with a given root box and k splits where $k \in \{i, i+1, \ldots, j\}$. ${}^{\circ}\mathbb{S}_{0:\infty} := \lim_{i\to\infty} {}^{\circ}\mathbb{S}_{0:i}$ is the space of all statistical regular pavings with a given root box.

Take a data sample of size n, $X_1, X_2, \ldots, X_n$ and an SRP ${}^{\circ}s$. For convenience the sample will be referred to as ${}^{n}X$. Let ${}^{\subset n}X$ be a subset of ${}^{n}X$ and let ${}^{\subset n}X_{\rho\mathsf{v}}$ be the subset of ${}^{n}X$ contained in the box $\boldsymbol{x}_{\rho\mathsf{v}}$ associated with a node $\rho\mathsf{v}$ in ${}^{\circ}s$.

A recursively computable statistic of some data is a statistic whose value can be updated for the addition of new data using only the current value of the statistic and the new data (i.e., it is not necessary to know the individual data values from which the current value of the statistic is calculated). Formally, if $T({}^{\subset n}X)$ is some statistic of ${}^{\subset n}X$ and a new data point x is added to ${}^{\subset n}X$ so that $n' = n + 1$ and ${}^{\subset n'}X = {}^{\subset n}X \cup x$, then $T({}^{\subset n'}X)$ can be calculated using $u(T({}^{\subset n}X), x)$ where u is some updating function.

Algorithm 3.1: `RPUnion`$(\rho\mathsf{v}^{(1)}, \rho\mathsf{v}^{(2)})$

---

input : Nodes $\rho\mathsf{v}^{(1)}$ and $\rho\mathsf{v}^{(2)}$ in RPs $s^{(1)}$ and $s^{(2)}$, respectively, with boxes $\boldsymbol{x}_{\rho\mathsf{v}^{(1)}} = \boldsymbol{x}_{\rho\mathsf{v}^{(2)}}$

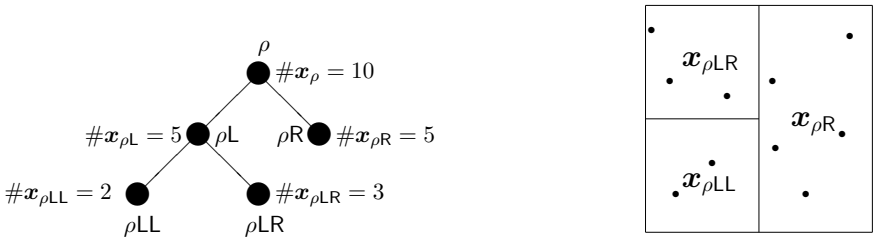
output : Node $\rho\mathsf{v}$ in RP $s = s^{(1)} \cup s^{(2)}$

Make a new node $\rho\mathsf{v}$
if `IsLeaf`$(\rho\mathsf{v}^{(1)})$ & `IsLeaf`$(\rho\mathsf{v}^{(2)})$ **then**
  $\rho\mathsf{v} \leftarrow$ `Copy`$(\rho\mathsf{v}^{(1)})$
  return $\rho\mathsf{v}$
end
else if !`IsLeaf`$(\rho\mathsf{v}^{(1)})$ & `IsLeaf`$(\rho\mathsf{v}^{(2)})$ **then**
  $\rho\mathsf{v} \leftarrow$ `Copy`$(\rho\mathsf{v}^{(1)})$
  return $\rho\mathsf{v}$
end
else if `IsLeaf`$(\rho\mathsf{v}^{(1)})$ & !`IsLeaf`$(\rho\mathsf{v}^{(2)})$ **then**
  $\rho\mathsf{v} \leftarrow$ `Copy`$(\rho\mathsf{v}^{(2)})$
  return $\rho\mathsf{v}$
end
else
  !`IsLeaf`$(\rho\mathsf{v}^{(1)})$ & !`IsLeaf`$(\rho\mathsf{v}^{(2)})$
end
$\boldsymbol{x}_{\rho\mathsf{v}} \leftarrow \boldsymbol{x}_{\rho\mathsf{v}^{(1)}}$
Graft onto $\rho\mathsf{v}$ as left child the node `RPUnion`$(\rho\mathsf{v}^{(1)}\mathsf{L}, \rho\mathsf{v}^{(2)}\mathsf{L})$
Graft onto $\rho\mathsf{v}$ as right child the node `RPUnion`$(\rho\mathsf{v}^{(1)}\mathsf{R}, \rho\mathsf{v}^{(2)}\mathsf{R})$
return $\rho\mathsf{v}$

The description above is deliberately vague about what recursively computable statistics are recorded by a node. In implementation terms (i.e., as implemented in computer code), different sub-types of SRP could be defined to record different sets of statistics chosen to suit the purpose for which the SRP is to be used.

For the purpose of this thesis, the only statistic that an SRP node $\rho\mathsf{v}$ is required to keep is the count of the number of data points in ${}^{\subset n}X_{\rho\mathsf{v}}$. This count is denoted by $\#\boldsymbol{x}_{\rho\mathsf{v}} = |{}^{\subset n}X_{\rho\mathsf{v}}|$. A leaf node $\rho\mathsf{v}$ with $\#\boldsymbol{x}_{\rho\mathsf{v}} > 0$ is a non-empty leaf node. The set of non-empty leaves of an SRP ${}^{\circ}s$ is $\mathbb{L}^{+}({}^{\circ}s) := \{\rho\mathsf{v} \in \mathbb{L}({}^{\circ}s) : \#\boldsymbol{x}_{\rho\mathsf{v}} > 0\} \subseteq \mathbb{L}({}^{\circ}s)$.

Figure 3.6 depicts a small SRP ${}^{\circ}s$ with root box $\boldsymbol{x}_{\rho} \in \mathbb{IR}^2$. The number of sample data points in the root box $\boldsymbol{x}_{\rho}$ is 10. Figure 3.6(a) shows the tree, including the count associated with each node in the tree. Figure 3.6(b) shows the partition of the root box represented by this tree, with the sample data points superimposed on the box.



(a) SRP tree. (b) SRP paving.

Figure 3.6: A small SRP.

The reason that an SRP is defined to record *recursively* computable statistics concerns computational efficiency. It is efficient to have an association between each leaf node and the actual data points falling inside the box of that leaf node because, if that node subsequently splits, *only* those data points can possibly be in the boxes associated with the new child nodes. If $\rho\mathsf{v}$ is a leaf node and the data in the box associated with $\rho\mathsf{v}$ is ${}^{\subset n}X_{\rho\mathsf{v}}$, an association between $\rho\mathsf{v}$ and ${}^{\subset n}X_{\rho\mathsf{v}}$ means that the data associated with the new left and right child nodes, ${}^{\subset n}X_{\rho\mathsf{vL}}$ and ${}^{\subset n}X_{\rho\mathsf{vL}}$, can be found by only considering the data points in ${}^{\subset n}X_{\rho\mathsf{v}}$. Without such an association all the *other* data points in ${}^{n}X$ but not in ${}^{\subset n}X_{\rho\mathsf{v}}$ would also need to be checked. The split operation $\bigtriangledown_{\text{SRP}}(\rho\mathsf{v})$ for SRP nodes is defined to incorporate this, as well as recomputation of the recursively computable statistics recorded by each node (see Algorithm 3.3).

It is, however, *not* necessary for the non-leaf (internal) nodes to be associated with data points in ${}^{n}X$, provided that the statistics recorded by each node are recursively computable and provided that the merge operation for SRP nodes is defined so that, after the operation, all the data points previously associated with the now-merged children are associated with

their parent, now a leaf node (see Algorithm 3.4). In terms of a computer implementation, this means that less memory is required to hold the total SRP structure.[1]

Let `GetAssociation`(x) be an operation that makes an association to a data point x.[2] An association to a single data point x is denoted by $\downarrow^x =$ `GetAssociation`(x). The set of associations between a node $\rho\mathsf{v}$ and the data points in ${}^{\subset n}X_{\rho\mathsf{v}}$ is denoted by $\rho\mathsf{v}_{[\downarrow]}$. If $\rho\mathsf{v}$ is not a leaf node, $\rho\mathsf{v}_{[\downarrow]} = \emptyset$. Let `GetData`$(\downarrow^x) = x$ be an operation that retrieves the data point $x \in {}^{n}X$ using the association $\downarrow^x$.

3.4.2 Operations on statistical regular pavings

Let `UpdateStatistics`$(\rho\mathsf{v}, x)$ be an operation that updates the recursively computable statistics associated with a node $\rho\mathsf{v}$ of an SRP for the addition of the data x to ${}^{\subset n}X_{\rho\mathsf{v}}$. The operation `InsertData` (Algorithm 3.2) tries to add a data point to an SRP ${}^{\circ}s$.

Algorithm 3.2: `InsertData` $(\rho\mathsf{v}, \downarrow^x)$

input : $\rho\mathsf{v}$ a node in an SRP ${}^{\circ}s$ and a data point association $\downarrow^x$
output: Boolean (**True** or **False**)

```
x ← GetData(↓^x)
if (x ∈ x_ρv) then
    UpdateStatistics(ρv, x)
    if IsLeaf(ρv) then
        ρv_[↓].append(↓^x)                          // append ↓^x to ρv_[↓]
    end
    else
        if !InsertData(ρvR, ↓^x) then
            InsertData(ρvL, ↓^x)
        end
    end
    return True
end
else
    return False
end
```

The tree structure means that `InsertData` is a $\mathcal{O}\left(\log\left(|\mathbb{L}({}^{\circ}s)|\right)\right)$ operation. `InsertData` can be used iteratively to try to add an entire data set ${}^{n}X$ to an SRP ${}^{\circ}s$.

The split and merge operations originally defined for RPs are extended for SRPs with $\bigtriangledown_{\text{SRP}}$ (Algorithm 3.3) and $\bigtriangleup_{\text{SRP}}$ (Algorithm 3.4). When a leaf node is split, its data is re-associated

[1] There is no reason, other than memory use, why data should not be associated with the non-leaf nodes: this issue is essentially an implementation detail except insofar as it affects the algorithms given in this thesis.

[2] Exactly how these associations are made is an implementation detail. The `C++` implementation used for this thesis uses pointers to a collection of data held outside the SRP structure; there are many other possible ways to achieve the same end.

with either its new left child node or its new right child node. The recursively computable statistics in each child node are updated for each data point associated with it. When two sibling nodes are merged, the data associations are moved up to the parent (whose recursively computable statistics remain unchanged) before the child nodes are removed.

Algorithm 3.3: $\bigtriangledown_{\text{SRP}}(\rho\mathsf{v})$

input : $\rho\mathsf{v}$ a node in an SRP $^{\circ}s$
output: $\{\rho\mathsf{vL}, \rho\mathsf{vR}\}$ left and right child nodes of $\rho\mathsf{v}$

$\bigtriangledown_{\text{RP}}(\rho\mathsf{v})$
foreach $\downarrow^x \in \rho\mathsf{v}_{[\downarrow]}$ **do**
  if ! `InsertData` $(\rho\mathsf{vR}, \downarrow^x)$ **then**
    `InsertData` $(\rho\mathsf{vL}, \downarrow^x)$
  end
end
$\rho\mathsf{v}_{[\downarrow]} \leftarrow \emptyset$ `// empty` $\rho\mathsf{v}_{[\downarrow]}$

Algorithm 3.4: $\bigtriangleup_{\text{SRP}}(\rho\mathsf{v})$

input : $\rho\mathsf{v} \in \mathbb{C}(^{\circ}s)$, a cherry node in an SRP $^{\circ}s$
input : $\rho\mathsf{v}$ such that $\rho\mathsf{v} \in \mathbb{C}(^{\circ}s)$, a leaf node in the SRP

$\rho\mathsf{v}_{[\downarrow]} \leftarrow \rho\mathsf{vL}_{[\downarrow]} \cup \rho\mathsf{vR}_{[\downarrow]}$
$\bigtriangleup_{\text{RP}}(\rho\mathsf{v})$

3.4.3 Statistical regular paving histograms

Given the count data recorded by each node, an SRP associated with data ${}^{n}X$ can be used to form a histogram. The bins are the elements in the partition, i.e., the boxes associated with the leaf nodes $\boldsymbol{x}_{\mathbb{L}(^{\circ}s)}$. If the total number of data points associated with the whole of an SRP $^{\circ}s$ with root node ρ and root box $\boldsymbol{x}_\rho$ is $n = \#\boldsymbol{x}_\rho = \sum_{\rho\mathsf{v}\in\mathbb{L}(^{\circ}s)} \#\boldsymbol{x}_{\rho\mathsf{v}}$, the equivalent of Equation (2.3) is:

$$\hat{f}_n(x) = \sum_{\rho\mathsf{v}\in\mathbb{L}(^{\circ}s)} \mathbb{1}_{\boldsymbol{x}_{\rho\mathsf{v}}}(x) \left(\frac{\#\boldsymbol{x}_{\rho\mathsf{v}}}{n\text{vol}\left(\boldsymbol{x}_{\rho\mathsf{v}}\right)} \right) \quad . \tag{3.3}$$

A histogram obtained using Equation (3.3) is referred to as an SRP histogram. Methods for finding an appropriate partition for an SRP histogram are discussed in Chapter 4.

SRP histograms have some similarities to dyadic histograms (Klemelä 2009, chap. 18). Both are binary tree-based and partition so that a box may only be bisected at the mid-point of one of its coordinates, but the RP structure restricts partitioning further by only bisecting a box on its first widest coordinate.

Given an SRP $^{\circ}s$ where the total number of data points associated with $^{\circ}s$ is n, the likelihood of the histogram is:

$$\mathcal{L}\left(\hat{f}_n\right) = \prod_{\rho\mathsf{v}\in\mathbb{L}(^{\circ}s)} \left(\frac{\#\boldsymbol{x}_{\rho\mathsf{v}}}{n\mathrm{vol}\left(\boldsymbol{x}_{\rho\mathsf{v}}\right)}\right)^{\#\boldsymbol{x}_{\rho\mathsf{v}}} . \tag{3.4}$$

and this is a maximum likelihood estimator over the class of simple (piecewise-constant) functions given the partition $\boldsymbol{x}_{\mathbb{L}(^{\circ}s)}$ of the root box of $^{\circ}s$ (Scott and Sain 2005).

The log-likelihood is given by

$$\begin{aligned}
\ell\left(\hat{f}_n\right) &= \log \mathcal{L}\left(\hat{f}_n\right) \\
&= \log\left(\prod_{\rho\mathsf{v}\in\mathbb{L}(^{\circ}s)} \left(\frac{\#\boldsymbol{x}_{\rho\mathsf{v}}}{n\mathrm{vol}\left(\boldsymbol{x}_{\rho\mathsf{v}}\right)}\right)^{\#\boldsymbol{x}_{\rho\mathsf{v}}}\right) \\
&= \sum_{\rho\mathsf{v}\in\mathbb{L}(^{\circ}s)} \#\boldsymbol{x}_{\rho\mathsf{v}} \log\left(\frac{\#\boldsymbol{x}_{\rho\mathsf{v}}}{n\mathrm{vol}\left(\boldsymbol{x}_{\rho\mathsf{v}}\right)}\right) \\
&= \sum_{\rho\mathsf{v}\in\mathbb{L}(^{\circ}s)} \#\boldsymbol{x}_{\rho\mathsf{v}} \left(\log\left(\#\boldsymbol{x}_{\rho\mathsf{v}}\right) - \log\left(n\mathrm{vol}\left(\boldsymbol{x}_{\rho\mathsf{v}}\right)\right)\right) \\
&= \sum_{\rho\mathsf{v}\in\mathbb{L}(^{\circ}s)} \#\boldsymbol{x}_{\rho\mathsf{v}} \log\left(\#\boldsymbol{x}_{\rho\mathsf{v}}\right) - \sum_{\rho\mathsf{v}\in\mathbb{L}(^{\circ}s)} \#\boldsymbol{x}_{\rho\mathsf{v}} \log\left(\mathrm{vol}\left(\boldsymbol{x}_{\rho\mathsf{v}}\right)\right) - \sum_{\rho\mathsf{v}\in\mathbb{L}(^{\circ}s)} \#\boldsymbol{x}_{\rho\mathsf{v}} \log\left(n\right) \\
&= \sum_{\rho\mathsf{v}\in\mathbb{L}(^{\circ}s)} \#\boldsymbol{x}_{\rho\mathsf{v}} \log\left(\#\boldsymbol{x}_{\rho\mathsf{v}}\right) - \sum_{\rho\mathsf{v}\in\mathbb{L}(^{\circ}s)} \#\boldsymbol{x}_{\rho\mathsf{v}} \log\left(\mathrm{vol}\left(\boldsymbol{x}_{\rho\mathsf{v}}\right)\right) - n\log\left(n\right) .
\end{aligned} \tag{3.5}$$

Recalling that the volume of a box is related to the depth of a node and the volume of the root box $\boldsymbol{x}_{\rho}$, the volume of the box $\boldsymbol{x}_{\rho\mathsf{v}}$ of a node $\rho\mathsf{v}$ with depth $\mathsf{d}_{\rho\mathsf{v}}$ is $\mathrm{vol}\left(\boldsymbol{x}_{\rho\mathsf{v}}\right) = \frac{\mathrm{vol}\left(\boldsymbol{x}_{\rho}\right)}{2^{\mathsf{d}_{\rho\mathsf{v}}}}$, and the log-likelihood can be expressed in terms of leaf node depths as

$$\begin{aligned}
\ell\left(\hat{f}_n\right) &= \sum_{\rho\mathsf{v}\in\mathbb{L}(^{\circ}s)} \#\boldsymbol{x}_{\rho\mathsf{v}} \log\left(\#\boldsymbol{x}_{\rho\mathsf{v}}\right) - \sum_{\rho\mathsf{v}\in\mathbb{L}(^{\circ}s)} \#\boldsymbol{x}_{\rho\mathsf{v}} \log\left(\mathrm{vol}\left(\boldsymbol{x}_{\rho\mathsf{v}}\right)\right) - n\log\left(n\right) \\
&= \sum_{\rho\mathsf{v}\in\mathbb{L}(^{\circ}s)} \#\boldsymbol{x}_{\rho\mathsf{v}} \log\left(\#\boldsymbol{x}_{\rho\mathsf{v}}\right) - \sum_{\rho\mathsf{v}\in\mathbb{L}(^{\circ}s)} \#\boldsymbol{x}_{\rho\mathsf{v}} \log\left(\frac{\mathrm{vol}\left(\boldsymbol{x}_{\rho}\right)}{2^{\mathsf{d}_{\rho\mathsf{v}}}}\right) - n\log\left(n\right) \\
&= \sum_{\rho\mathsf{v}\in\mathbb{L}(^{\circ}s)} \#\boldsymbol{x}_{\rho\mathsf{v}} \log\left(\#\boldsymbol{x}_{\rho\mathsf{v}}\right) + \sum_{\rho\mathsf{v}\in\mathbb{L}(^{\circ}s)} \mathsf{d}_{\rho\mathsf{v}}\#\boldsymbol{x}_{\rho\mathsf{v}} \log\left(2\right) - \sum_{\rho\mathsf{v}\in\mathbb{L}(^{\circ}s)} \#\boldsymbol{x}_{\rho\mathsf{v}} \log\left(\mathrm{vol}\left(\boldsymbol{x}_{\rho}\right)\right) - n\log\left(n\right) \\
&= \sum_{\rho\mathsf{v}\in\mathbb{L}(^{\circ}s)} \#\boldsymbol{x}_{\rho\mathsf{v}} \log\left(\#\boldsymbol{x}_{\rho\mathsf{v}}\right) + \log\left(2\right) \sum_{\rho\mathsf{v}\in\mathbb{L}(^{\circ}s)} \mathsf{d}_{\rho\mathsf{v}}\#\boldsymbol{x}_{\rho\mathsf{v}} - \log\left(\mathrm{vol}\left(\boldsymbol{x}_{\rho}\right)\right) \sum_{\rho\mathsf{v}\in\mathbb{L}(^{\circ}s)} \#\boldsymbol{x}_{\rho\mathsf{v}} - n\log\left(n\right) \\
&= \left(\sum_{\rho\mathsf{v}\in\mathbb{L}(^{\circ}s)} \#\boldsymbol{x}_{\rho\mathsf{v}} \log\left(\#\boldsymbol{x}_{\rho\mathsf{v}}\right)\right) + \left(\log\left(2\right) \sum_{\rho\mathsf{v}\in\mathbb{L}(^{\circ}s)} \mathsf{d}_{\rho\mathsf{v}}\#\boldsymbol{x}_{\rho\mathsf{v}}\right) - n\log\left(\mathrm{vol}\left(\boldsymbol{x}_{\rho}\right)\right) - n\log\left(n\right) .
\end{aligned} \tag{3.6}$$

If the partition is changed by splitting a single leaf node $\rho\mathsf{v}$ to form two child nodes $\rho\mathsf{vL}$

and $\rho\mathsf{vR}$, then the change in the log-likelihood $\delta\ell_{\bigtriangledown_{\mathrm{SRP}}(\rho\mathsf{v})}$ is

$$\delta\ell_{\bigtriangledown_{\mathrm{SRP}}(\rho\mathsf{v})} = \#\boldsymbol{x}_{\rho\mathsf{vL}} \log\left(\#\boldsymbol{x}_{\rho\mathsf{vL}}\right) + \#\boldsymbol{x}_{\rho\mathsf{vR}} \log\left(\#\boldsymbol{x}_{\rho\mathsf{vR}}\right) - \#\boldsymbol{x}_{\rho\mathsf{v}} \log\left(\#\boldsymbol{x}_{\rho\mathsf{v}}\right) + \#\boldsymbol{x}_{\rho\mathsf{v}} \log\left(2\right) \quad . \tag{3.7}$$

Similarly, if the partition is changed by merging the children $\rho\mathsf{vL}$ and $\rho\mathsf{vR}$ of a single cherry node $\rho\mathsf{v}$ back into $\rho\mathsf{v}$, then the change in the log-likelihood $\delta\ell_{\bigtriangleup_{\mathrm{SRP}}(\rho\mathsf{v})}$ is

$$\delta\ell_{\bigtriangleup_{\mathrm{SRP}}(\rho\mathsf{v})} = \#\boldsymbol{x}_{\rho\mathsf{v}} \log\left(\#\boldsymbol{x}_{\rho\mathsf{v}}\right) - \left(\#\boldsymbol{x}_{\rho\mathsf{vL}} \log\left(\#\boldsymbol{x}_{\rho\mathsf{vL}}\right) + \#\boldsymbol{x}_{\rho\mathsf{vR}} \log\left(\#\boldsymbol{x}_{\rho\mathsf{vR}}\right)\right) - \#\boldsymbol{x}_{\rho\mathsf{v}} \log\left(2\right) \quad . \tag{3.8}$$

3.5 Mapped regular pavings

3.5.1 Definition

A mapped regular paving (MRP) is another extension of an RP. Let $s \in \mathbb{S}_{0:\infty}$ be an RP with root node ρ and root box $\boldsymbol{x}_\rho \in \mathbb{IR}^d$ and let $\mathbb{Y}$ be a non-empty set. Let ${}^{\blacksquare}f : \mathbb{V}(s) \to \mathbb{Y}$ map each node of s to an element in $\mathbb{Y}$ as follows:

$$\{\rho\mathsf{v} \mapsto f_{\rho\mathsf{v}} : \rho\mathsf{v} \in \mathbb{V}(s), f_{\rho\mathsf{v}} \in \mathbb{Y}\} \quad .$$

Such a map ${}^{\blacksquare}f$ is called a $\mathbb{Y}$-*mapped regular paving* ($\mathbb{Y}$-MRP). Thus, a $\mathbb{Y}$-MRP ${}^{\blacksquare}f$ is obtained by augmenting each node $\rho\mathsf{v}$ of the RP tree s with an additional data member $f_{\rho\mathsf{v}} \in \mathbb{Y}$.

The sets of all nodes and leaf nodes of an MRP ${}^{\blacksquare}f$ are denoted by $\mathbb{V}({}^{\blacksquare}f)$ and $\mathbb{L}({}^{\blacksquare}f)$, respectively. The set of all leaf node boxes is denoted by $\boldsymbol{x}_{\mathbb{L}({}^{\blacksquare}f)}$

Let the class of $\mathbb{Y}$-MRPs over the leaf boxes of regular pavings of a root box $\boldsymbol{x}_\rho \in \mathbb{IR}^d$ be :

$${}^{\blacksquare}\mathcal{F} := \{\{\rho\mathsf{v} \mapsto f_{\rho\mathsf{v}} : \rho\mathsf{v} \in \mathbb{V}(s), f_{\rho\mathsf{v}} \in \mathbb{Y}\} : s \in \mathbb{S}_{0:\infty}\}$$

3.5.2 Operations on mapped regular pavings

Let $\eta(x)$ be the leaf node of a $\mathbb{Y}$-MRP ${}^{\blacksquare}f$ whose box $\boldsymbol{x}_{\eta(x)}$ in the set of all leaf boxes $\boldsymbol{x}_{\mathbb{L}({}^{\blacksquare}f)}$ contains the point $x \in \boldsymbol{x}_\rho \in \mathbb{IR}^d$. Operation `MRPPointwiseImage` (Algorithm 3.5) locates $\eta(x)$ and returns $f_{\eta(x)}$. Like `InsertData`, `MRPPointwiseImage` is a $\mathcal{O}\left(\log\left(|\mathbb{L}({}^{\blacksquare}f)|\right)\right)$ operation because of the tree structure of the MRP.

`MRPPointwiseImage` can be used to give the *pointwise extension* $f : \boldsymbol{x}_\rho \to \mathbb{Y}$ given by the piecewise constant map $x \mapsto f_{\eta(x)}$. Define

$${}^{\blacksquare}f(x) := \texttt{MRPPointwiseImage}(\rho, x) \tag{3.9}$$

where ρ is the root node of ${}^{\blacksquare}f$.

Algorithm 3.5: `MRPPointwiseImage`$(\rho\mathsf{v}, x)$

input : $\rho\mathsf{v}$ with box $\boldsymbol{x}_{\rho\mathsf{v}}$, a node in a $\mathbb{Y}$-MRP ${}^{\blacksquare}f$, and a point $x \in \boldsymbol{x}_{\rho\mathsf{v}}$.
output : Return $f_{\eta(x)}$ for the leaf node $\eta(x)$ descended from $\rho\mathsf{v}$ whose box $\boldsymbol{x}_{\eta(x)}$ contains x.

```
if IsLeaf(ρv) then
  return f_ρv
end
else
  if x ∈ x_ρvR then
    MRPPointWiseImage(ρvR, x)
  end
  else
    MRPPointWiseImage(ρvL, x)
  end
end
```

This pointwise extension can be used to extend arithmetic from $\mathbb{Y}$ to ${}^{\blacksquare}\mathcal{F}$. Let ${}^{\blacksquare}f$ and ${}^{\blacksquare}g$ be two $\mathbb{Y}$-MRPs with the same root box $\boldsymbol{x}_\rho$ and let $\mathbb{Y}$ be a field: extending arithmetic from $\mathbb{Y}$ to ${}^{\blacksquare}\mathcal{F}$ means that $({}^{\blacksquare}f \star {}^{\blacksquare}g)(x) = {}^{\blacksquare}f(x) \star {}^{\blacksquare}g(x)$, where $\star \in \{+, -, \cdot, /\}$, up to the operation $\star$ being well-defined in $\mathbb{Y}$ for every $x \in \boldsymbol{x}_\rho$.

The additive and multiplicative identities, say $\mathbb{Y}(0)$ and $\mathbb{Y}(1)$ in the field $\mathbb{Y}$, can be used to get the additive and multiplicative identities in ${}^{\blacksquare}\mathcal{F}$, ${}^{\blacksquare 0}f$ and ${}^{\blacksquare 1}f$ respectively:

- ${}^{\blacksquare 0}f$ is the constant $\mathbb{Y}$-MRP such that ${}^{\blacksquare 0}f(x) = \mathbb{Y}(0)\ \forall x \in \boldsymbol{x}_\rho$ the root box of ${}^{\blacksquare 0}f$.
- ${}^{\blacksquare 1}f$ is the constant $\mathbb{Y}$-MRP such that ${}^{\blacksquare 1}f(x) = \mathbb{Y}(1)\ \forall x \in \boldsymbol{x}_\rho$ the root box of ${}^{\blacksquare 1}f$.

Operation `MRPOperate` (Algorithm 3.6) is an extension of `RPUnion` (Algorithm 3.1) used to perform binary operations over nodes in $\mathbb{Y}$-MRPs. Note that, like `RPUnion`, `MRPOperate` can be expressed very simply because it recurses simultaneously on pairs of nodes and the pairs are easily found as a result of the tree structure of the pavings. Also like `RPUnion`, `MRPOperate` can be used on any two $\mathbb{Y}$-MRP nodes subject only to the condition that they have the same root box because of the restrictive bisection rule used to create a regular paving.

Define the operation `MRPTransform`$(\rho\mathsf{v}, \tau)$ to apply the unary transformation $\tau : \mathbb{Y} \to \mathbb{Y}$ to a given node $\rho\mathsf{v}$ in a $\mathbb{Y}$-MRP ${}^{\blacksquare}f$ by recursively descending through the tree and setting $f_{\rho\mathsf{v}} \leftarrow \tau(f_{\rho\mathsf{v}})$ for $\rho\mathsf{v}$ and its descendants.

Define the operation `MRPTransform`$(\rho\mathsf{v}, y, \star)$, $y \in \mathbb{Y}$, to apply a binary transformation $\star : (\mathbb{Y}, \mathbb{Y}) \to \mathbb{Y}$ to a given node $\rho\mathsf{v}$ in a $\mathbb{Y}$-MRP ${}^{\blacksquare}f$ by recursively descending through the tree and setting $f_{\rho\mathsf{v}} \leftarrow f_{\rho\mathsf{v}} \star y$ for $\rho\mathsf{v}$ and its descendants, again up to the operation $\star$ being well-defined in $\mathbb{Y}$.

Algorithm 3.6: $\texttt{MRPOperate}(\rho\mathsf{v}^{(f)}, \rho\mathsf{v}^{(g)}, \star)$

input : nodes $\rho\mathsf{v}^{(f)}$ from ${}^{\blacksquare}f$ and $\rho\mathsf{v}^{(f)}$ from ${}^{\blacksquare}g$ with same box $\boldsymbol{x}_{\rho\mathsf{v}^{(f)}} = \boldsymbol{x}_{\rho\mathsf{v}^{(g)}}$ and binary operation $\star$.

output : node $\rho\mathsf{v}$ of $\mathbb{Y}$-MRP ${}^{\blacksquare}h = {}^{\blacksquare}f \star {}^{\blacksquare}g$.

Make a new node $\rho\mathsf{v}$ with box and image
$\boldsymbol{x}_{\rho\mathsf{v}} \leftarrow \boldsymbol{x}_{\rho\mathsf{v}^{(f)}}$; $h_{\rho\mathsf{v}} \leftarrow f_{\rho\mathsf{v}^{(f)}} \star g_{\rho\mathsf{v}^{(g)}}$

if $\texttt{IsLeaf}(\rho\mathsf{v}^{(f)})$ & $!\,\texttt{IsLeaf}(\rho\mathsf{v}^{(g)})$ **then**
  Make nodes L', R'
  $\boldsymbol{x}_{\mathsf{L}'} \leftarrow \boldsymbol{x}_{\rho\mathsf{v}^{(g)}\mathsf{L}}$; $\boldsymbol{x}_{\mathsf{R}'} \leftarrow \boldsymbol{x}_{\rho\mathsf{v}^{(g)}\mathsf{R}}$
  $f_{\mathsf{L}'} \leftarrow f_{\rho\mathsf{v}^{(f)}}$, $f_{\mathsf{R}'} \leftarrow f_{\rho\mathsf{v}^{(f)}}$
  Graft onto $\rho\mathsf{v}$ as left child the node $\texttt{MRPOperate}(\mathsf{L}', \rho\mathsf{v}^{(g)}\mathsf{L}, \star)$
  Graft onto $\rho\mathsf{v}$ as right child the node $\texttt{MRPOperate}(\mathsf{R}', \rho\mathsf{v}^{(g)}\mathsf{R}, \star)$
end

else if $!\,\texttt{IsLeaf}(\rho\mathsf{v}^{(f)})$ & $\texttt{IsLeaf}(\rho\mathsf{v}^{(g)})$ **then**
  Make nodes L', R'
  $\boldsymbol{x}_{\mathsf{L}'} \leftarrow \boldsymbol{x}_{\rho\mathsf{v}^{(f)}\mathsf{L}}$; $\boldsymbol{x}_{\mathsf{R}'} \leftarrow \boldsymbol{x}_{\rho\mathsf{v}^{(f)}\mathsf{R}}$
  $g_{\mathsf{L}'} \leftarrow f_{\rho\mathsf{v}^{(g)}}$, $g_{\mathsf{R}'} \leftarrow f_{\rho\mathsf{v}^{(g)}}$
  Graft onto $\rho\mathsf{v}$ as left child the node $\texttt{MRPOperate}(\rho\mathsf{v}^{(f)}\mathsf{L}, \mathsf{L}', \star)$
  Graft onto $\rho\mathsf{v}$ as right child the node $\texttt{MRPOperate}(\rho\mathsf{v}^{(f)}\mathsf{R}, \mathsf{R}', \star)$
end

else if $!\,\texttt{IsLeaf}(\rho\mathsf{v}^{(f)})$ & $!\,\texttt{IsLeaf}(\rho\mathsf{v}^{(g)})$ **then**
  Graft onto $\rho\mathsf{v}$ as left child the node $\texttt{MRPOperate}(\rho\mathsf{v}^{(f)}\mathsf{L}, \rho\mathsf{v}^{(g)}\mathsf{L}, \star)$
  Graft onto $\rho\mathsf{v}$ as right child the node $\texttt{MRPOperate}(\rho\mathsf{v}^{(f)}\mathsf{R}, \rho\mathsf{v}^{(g)}\mathsf{R}, \star)$
end
return $\rho\mathsf{v}$

Let ${}^{\blacksquare}f$ and ${}^{\blacksquare}g$ be two $\mathbb{Y}$-MRPs with the same root box $\boldsymbol{x}_\rho$ and root nodes $\rho^{(f)}$, $\rho^{(g)}$, respectively. Arithmetic on $\mathbb{Y}$-MRPs can be defined by defining:

$$ {}^{\blacksquare}f \star {}^{\blacksquare}g := \texttt{MRPOperate}(\rho^{(f)}, \rho^{(g)}, \star) \tag{3.10a} $$

$$ \tau({}^{\blacksquare}f) := \texttt{MRPTransform}(\rho^{(f)}, \tau) \tag{3.10b} $$

$$ {}^{\blacksquare}f \star y := \texttt{MRPTransform}(\rho^{(f)}, y, \star) \quad \text{for } y \in \mathbb{Y} \tag{3.10c} $$

Operation `MRPSlice` (Algorithm 3.7) 'slices' a $\mathbb{Y}$-MRP ${}^{\blacksquare}f$ orthogonal to one or more of its coordinates. If the $\mathbb{Y}$-MRP to be sliced is ${}^{\blacksquare}f : \boldsymbol{x}_\rho \to \mathbb{Y}$ with root box $\boldsymbol{x}_\rho \in \mathbb{IR}^d$, let $\Delta = \{1, 2, \ldots, d\}$ and let $\Lambda \subset \Delta$ such that $\Lambda \neq \emptyset$. `MRPSlice` uses the following notation for tuples and subtuples of a point or Δ-tuple:

$$ x = (x_1, \ldots, x_d) =: (x_i)_{i\in\Delta} \in \boldsymbol{x}_\rho, \text{ and } \boldsymbol{x}_\rho = (\boldsymbol{x}_{\rho,i})_{i\in\Delta} = \bigboxtimes_{i\in\Delta} [\underline{x}_{\rho,i}, \overline{x}_{\rho,i}] \in \mathbb{IR}^d \ . $$

The Λ-subtuple of a Δ-tuple is obtained by retaining the subset of coordinates Λ out of the d coordinates in Δ as follows:

$$(x_j)_{j\in\Lambda} \in (\boldsymbol{x}_{\rho,j})_{j\in\Lambda} = \bigboxtimes_{j\in\Lambda} [\underline{x}_{\rho,j}, \overline{x}_{\rho,j}] \in \mathbb{IR}^{|\Lambda|} \ .$$

A 'slice' of the $\mathbb{Y}$-MRP ${}^{\blacksquare}f$ at a fixed subtuple point $x^\dagger = (x_j^\dagger)_{j\in\Lambda} \in (\boldsymbol{x}_{\rho,j})_{j\in\Lambda}$ is the $\mathbb{Y}$-MRP ${}^{\blacksquare}f^{|x^\dagger}$ with root box $(\boldsymbol{x}_{\rho,i})_{i\in\Delta\setminus\Lambda} := \bigboxtimes_{i\in\Delta\setminus\Lambda} [\underline{x}_{\rho,i}, \overline{x}_{\rho,i}]$ that satisfies:

$$ {}^{\blacksquare}f^{|x^\dagger}\left((x_i)_{i\in\Delta\setminus\Lambda}\right) = {}^{\blacksquare}f\left((x_i)_{i\in\Delta}\right), \quad \forall\, (x_i)_{i\in\Delta\setminus\Lambda} \in (\boldsymbol{x}_{\rho,i})_{i\in\Delta\setminus\Lambda}\,, \quad \text{whenever } (x_i)_{i\in\Lambda} = (x^\dagger_{j=i})_{j\in\Lambda} \tag{3.11}$$

The box $(\boldsymbol{x}_{\rho\mathsf{v},i})_{i\in\Delta\setminus\Lambda}$ associated with each node $\rho\mathsf{v}$ of ${}^{\blacksquare}f^{|x^\dagger}$ is such that there is a unique node $\rho\mathsf{u}$ in ${}^{\blacksquare}f$ such that $(\boldsymbol{x}_{\rho\mathsf{u},i})_{i\in\Delta\setminus\Lambda} = (\boldsymbol{x}_{\rho\mathsf{v},i})_{i\in\Delta\setminus\Lambda}$ and $x^\dagger = (x_j^\dagger)_{j\in\Lambda} \in (\boldsymbol{x}_{\rho\mathsf{u},j})_{j\in\Lambda}$. Equation (3.11) can be satisfied by then ensuring that $f^{|x^\dagger}_{\rho\mathsf{v}} = f_{\rho\mathsf{u}}$.

Algorithm 3.7: `MRPSlice`$(\rho\mathsf{v}, \Lambda, x^\dagger)$

input : $\rho\mathsf{v}$, a node in a $\mathbb{Y}$-MRP ${}^{\blacksquare}f$ with box
$\boldsymbol{x}_{\rho\mathsf{v}} = (\boldsymbol{x}_{\rho\mathsf{v},j})_{j\in\Delta} = \bigboxtimes_{j\in\Delta} [\underline{x}_{\rho\mathsf{v},j}, \overline{x}_{\rho\mathsf{v},j}] \in \mathbb{IR}^d$,
$\Lambda \subset \Delta$, such that $0 < |\Lambda| < |\Delta| = d$,
Point $x^\dagger = (x_j^\dagger)_{j\in\Lambda} \in (\boldsymbol{x}_{\rho\mathsf{v},j})_{j\in\Lambda}$

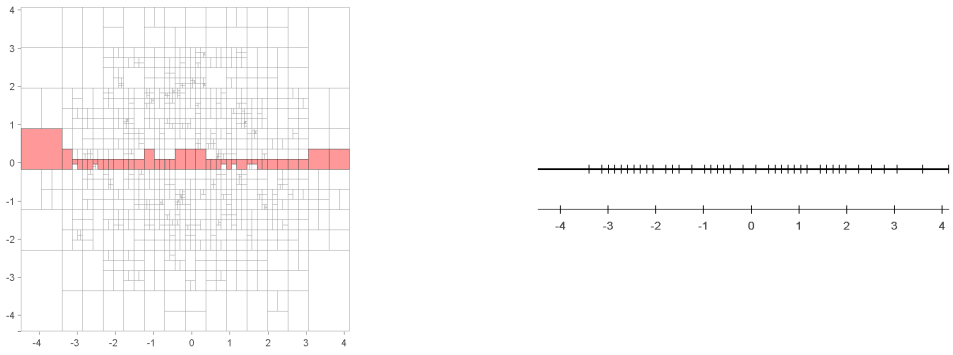
output : Change $\rho\mathsf{v}$ into a node in a $\mathbb{Y}$-MRP ${}^{\blacksquare}f^{|x^\dagger}$ that satisfies (3.11).

```
if IsLeaf(ρv) Or ι ∉ Λ then
    if !IsLeaf(ρv) then
        MRPSlice(ρvL, Λ, x†)
        MRPSlice(ρvR, Λ, x†)
    end
    x_ρv ← x'_ρv = ⊠_{j∈Δ\Λ} [x_ρv,j, x̄_ρv,j]        // f_ρv unchanged
end
else
    if x†_ι < mid [x_ι, x̄_ι] then
        MRPSlice(ρvL, Λ, x†)
        ρv ← Copy(ρvL)
    end
    else
        MRPSlice(ρvR, Λ, x†)
        ρv ← Copy(ρvR)
    end
end
```

Algorithm 3.7 uses the definitions given in Section 3.3.1 for the bisection of a box. ι is the

first coordinate of maximum width of a box $\boldsymbol{x}$ and $\operatorname{mid}[\underline{x}_\iota, \overline{x}_\iota]$ is the mid-point of the interval $[\underline{x}_\iota, \overline{x}_\iota]$. A bisection gives the left child a box where the interval on coordinate ι is open at the top. Thus if $x_\iota^\dagger < \operatorname{mid}[\underline{x}_\iota, \overline{x}_\iota]$, $x_\iota^\dagger$ is considered to be in the left child $\rho\mathsf{L}$'s box, otherwise $x_\iota^\dagger$ is considered to be in the right child $\rho\mathsf{R}$'s box. Algorithm 3.7 can be used to obtain a slice $\mathbb{Y}$-MRP of a given $\mathbb{Y}$-MRP at a specified subtuple $x^\dagger$.

Figure 3.7 illustrates the effect of Algorithm 3.7. Figure 3.7(a) shows, shaded, the boxes of the leaf nodes of an MRP ${}^{\blacksquare}f$ with root box $\boldsymbol{x}_\rho \in \mathbb{IR}^2$ that would be used to make the boxes of the leaf nodes of a slice ${}^{\blacksquare}f^{|x_2=0}$. The slice has root box $\boldsymbol{x}_\rho^{\downarrow} \in \mathbb{R}$ and each leaf node has a box which is just the interval on coordinate 1 of the shaded boxes in Figure 3.7(a). Figure 3.7(b) shows the partition of the total root $\boldsymbol{x}_\rho^{\downarrow}$ into these $1-d$ boxes. Each node $\rho\mathsf{v}$ in the slice ${}^{\blacksquare}f^{|x_2=0}$ will have the same value $f_{\rho\mathsf{v}}$ mapped to it as the node of ${}^{\blacksquare}f$ whose box is used to form the box $\boldsymbol{x}_{\rho\mathsf{v}}$ associated with $\rho\mathsf{v}$.



(a) Partition of ${}^{\blacksquare}f$ with boxes used to make slice shown shaded.

(b) Partition of the slice ${}^{\blacksquare}f^{|x_2=0}$ of ${}^{\blacksquare}f$.

Figure 3.7: Partitions showing the effect of slicing to create ${}^{\blacksquare}f^{|x_2=0}$.

3.6 Real-mapped regular pavings as piecewise-constant function estimates

3.6.1 Definition

A real-mapped regular paving (RMRP) is an $\mathbb{R}$-MRP. Arithmetic operations in $\mathbb{R}$ can be extended to RMRPs using the definitions in Section 3.5.2. An RMRP is denoted by ${}^{\square}f$. The sets of all nodes and leaf nodes of an RMRP ${}^{\square}f$ are denoted by $\mathbb{V}({}^{\square}f)$ and $\mathbb{L}({}^{\square}f)$, respectively. The set of all leaf node boxes is denoted by $\boldsymbol{x}_{\mathbb{L}({}^{\square}f)}$. Let the class of RMRPs over the leaf boxes of regular pavings of a root box $\boldsymbol{x}_\rho \in \mathbb{IR}^d$ be:

$$ {}^{\square}\mathcal{F} := \{\{\rho\mathsf{v} \mapsto f_{\rho\mathsf{v}} : \rho\mathsf{v} \in \mathbb{V}(s), f_{\rho\mathsf{v}} \in \mathbb{R}\} : s \in \mathbb{S}_{0:\infty}\} $$

Given any two RMRPs ${}^{\square}f^{(1)}$ and ${}^{\square}f^{(2)}$ with the same root box $\boldsymbol{x}_\rho$ and a binary operation $\star \in \{+, -, \cdot, /\}$, the RMRP ${}^{\square}f = {}^{\square}f^{(1)} \star {}^{\square}f^{(2)}$ can be obtained using Equation (3.10a)). An RMRP ${}^{\square}f$ can also be transformed using any standard function $\tau \in \mathfrak{S} := \{\exp, \sin, \cos, \tan, \ldots\}$ to obtain the RMRP $\tau({}^{\square}f)$ (Equation (3.10b)). Finally, a binary operation of the form ${}^{\square}f \star x$ for an RMRP ${}^{\square}f$ and $x \in \mathbb{R}$ can also be carried out (Equation (3.10c)), and again the result ${}^{\square}g = {}^{\square}f \star x$ is an RMRP.

RMRPs are important structures in this thesis because an RMRP can be used to represent a piecewise-constant function (PCF). Harlow et al. (2012) describes function approximation using RMRPs in general. This thesis is primarily concerned with RMRPs as nonparametric density estimates rather than as approximations to known functions, but the ability to create RMRP-structured function approximations is still highly relevant. This is explored further in Chapter 7.

The advantage of representing an RMRP-representation is that all the arithmetic operations in $\mathbb{R}$ described above can be carried out on RMRP-represented PCFs. In fact, the range of operations possible using RMRPs is even wider. A box in any type of RP has real volume $(\mathrm{vol}\,(\boldsymbol{x}_{\rho\mathsf{v}}) \in \mathbb{R})$. In general binary operations of the form $y \star x$ or $x \star y$ may not be defined for $y \in \mathbb{Y}$, $x \in \mathbb{R}$, but these operations are defined when $\mathbb{Y} = \mathbb{R}$. This allows operations using both node volume and node value, such as integrating, normalising and marginalising, to be carried out on RMRPs.

A *non-negative* RMRP ${}^{\square}f$ can be used to represent a (possibly non-normalised) probability density function. An RMRP ${}^{\square}f$ is non-negative if $f_{\rho\mathsf{v}} \geq 0 \ \forall \rho\mathsf{v} \in \mathbb{L}({}^{\square}f)$. Figure 3.8 shows an RMRP density estimate of an example density, Density I in Appendix B, with $d = 2$. This example is discussed further in Chapter 6.

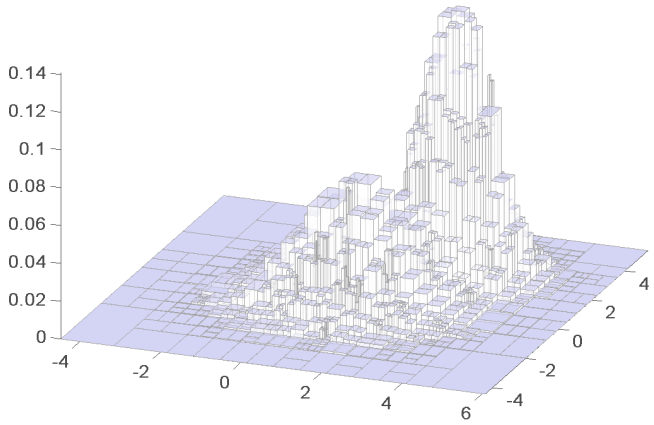


Figure 3.8: RMRP density estimate of Density I, $d = 2$.

Section 3.6.2 describes how the following RMRP operations can be performed:

- Integrating an RMRP;
- Normalising an RMRP;
- Averaging RMRPs to get another RMRP;
- Obtaining the pointwise image, under the RMRP, of a point in the RMRP domain;
- Marginalising a multivariate RMRP;
- Obtaining a conditional RMRP from a multivariate RMRP;
- Multiplying RMRPs to get another RMRP;
- Finding the L_1 distance between two RMRPs;
- Finding the highest coverage regions of an RMRP;
- Simulating data from an RMRP;

Section 3.7 describes how an SRP can be used to create a histogram represented by an RMRP.

3.6.2 Operations on piecewise-constant function estimates

3.6.2.1 Integrating

Let the set of boxes of all nodes on an RMRP ${}^{\square}f$ be $\boldsymbol{x}_{\mathbb{V}({}^{\square}f)}$. Operation `Integral` (Algorithm 3.8) shows how the integral of an RMRP ${}^{\square}f$ over some box $\boldsymbol{x} \in \boldsymbol{x}_{\mathbb{V}({}^{\square}f)}$ can be obtained. `Integral` can be used to define integration on ${}^{\square}\mathcal{F}$. Given an RMRP ${}^{\square}f$ with node boxes $\boldsymbol{x}_{\mathbb{V}({}^{\square}f)}$ and a box $\boldsymbol{x} \in \boldsymbol{x}_{\mathbb{V}({}^{\square}f)}$, define

$$\int_{\boldsymbol{x}} {}^{\square}f := \texttt{Integral}(\rho, \boldsymbol{x}) \ .$$

Thus `Integral` can be used to obtain the total integral of an RMRP ${}^{\square}f$ over its root box $\boldsymbol{x}_\rho$ as $\int_{\boldsymbol{x}_\rho} {}^{\square}f$.

3.6.2.2 Normalising

Consider a non-negative RMRP ${}^{\square}f$ with $f_{\rho\mathsf{v}} > 0$ for at least one $\rho\mathsf{v} \in \mathbb{L}({}^{\square}f)$. Such an RMRP is termed a *positive* RMRP. If ${}^{\square}f$ is a positive RMRP with root node ρ and root box $\boldsymbol{x}_\rho$ then $\texttt{Integral}(\rho, \boldsymbol{x}_\rho) > 0$. Division of an RMRP ${}^{\square}f$ by a real value $c \neq 0 \in \mathbb{R}$ is defined (Equation (3.10c)), and ${}^{\square}g = \frac{{}^{\square}f}{c}$ is also an RMRP. Thus a positive RMRP ${}^{\square}f$ with root node ρ and root box $\boldsymbol{x}_\rho$ can be normalised so that $\int_{\boldsymbol{x}_\rho} {}^{\square}f = 1$ using operation `Normalise` (Algorithm 3.9).

Algorithm 3.8: `Integral`$(\rho\mathsf{v}, \boldsymbol{x})$

input : $\rho\mathsf{v}$, a node in an RMRP ${}^{\square}f$, with root box $\boldsymbol{x}_\rho$ and $\boldsymbol{x} \in \boldsymbol{x}_{\mathbb{V}({}^{\square}f)}$
output : $\mathcal{I} \in \mathbb{R}$, the integral of ${}^{\square}f$ over $\boldsymbol{x}$

initialize: $\mathcal{I} \leftarrow 0$

if $x_{\rho\mathsf{v}} \not\subseteq \boldsymbol{x}$ **then**
  if $\boldsymbol{x} \subseteq \square\boldsymbol{x}_{\rho\mathsf{vR}}$ **then**
    $\mathcal{I} \leftarrow$ `Integral`$(\rho\mathsf{vR}, \boldsymbol{x})$
  end
  else
    $\mathcal{I} \leftarrow$ `Integral`$(\rho\mathsf{vL}, \boldsymbol{x})$
  end
end
else
  if `IsLeaf`$(\rho\mathsf{v})$ **then**
    $\mathcal{I} \leftarrow f_{\rho\mathsf{v}}\text{vol}\ (x_{\rho\mathsf{v}})$
  end
  else
    $\mathcal{I} \leftarrow$ `Integral`$(\rho\mathsf{vL}, \boldsymbol{x})$ + `Integral`$(\rho\mathsf{vR}, \boldsymbol{x})$
  end
end
return $\mathcal{I}$

Algorithm 3.9: `Normalise`$({}^{\square}f)$

input : Positive RMRP ${}^{\square}f$ with root box $\boldsymbol{x}_\rho$
output : ${}^{\square}f$ normalised so that $\int_{\boldsymbol{x}_\rho} {}^{\square}f = 1$

${}^{\square}f \leftarrow \dfrac{{}^{\square}f}{\texttt{Integral}(\rho, \boldsymbol{x}_\rho)}$ `// Using Equation` (3.10c)

return ${}^{\square}f$

3.6.2.3 Averaging

Two RMRPs ${}^{\square}f^{(1)}$ and ${}^{\square}f^{(2)}$ with the same root box $\boldsymbol{x}_\rho$ can be added together to obtain RMRP ${}^{\square}f = {}^{\square}f^{(1)} + {}^{\square}f^{(2)}$ (Equation (3.10a)). Another RMRP ${}^{\square}f^{(3)}$ with the same root box can then be added to ${}^{\square}f$, etc. Hence any finite number N of RMRPs can be added together to give another RMRP provided that they all have the same root box.

Division by $N \in \mathbb{R}$ is also defined (Equation (3.10c)) and ${}^{\square}g = \frac{{}^{\square}f}{N}$ is also an RMRP. Thus given a collection of N RMRPs ${}^{\square}f_1, \ldots, {}^{\square}f_N$, Equations (3.10a) and (3.10a) can be used to get the average RMRP ${}^{\square}\bar{f}_N$:

$$ {}^{\square}\bar{f}_N = \frac{\sum_{i=1}^{N} {}^{\square}f_i}{N} \tag{3.12} $$

If each ${}^{\square}f_i$ averaged represents a normalised probability density function then the average ${}^{\square}\bar{f}_N$ will also be non-negative and will integrate to 1.

3.6.2.4 Pointwise image

Given an RMRP ${}^{\square}f$ with root box $\boldsymbol{x}_\rho$, Equation (3.9) can be used to look up the image ${}^{\square}f(x)$ under the RMRP of any point $x \in \boldsymbol{x}_\rho$.

3.6.2.5 Marginalising

The same notation for tuples and subtuples is used here as was used to define operation `MRPSlice` (Algorithm 3.7). The marginal function ${}^{\square}f^{\Lambda}$ with root box $\boldsymbol{x}_\rho^{\Lambda}$ of an RMRP ${}^{\square}f$ with root box $\boldsymbol{x}_\rho = \boxtimes_{j\in\Delta}[\underline{x}_{\rho,j}, \overline{x}_{\rho,j}] \in \mathbb{IR}^d$, $d > 1$, is given by:

$$ {}^{\square}f^{\Lambda}\left((x_i)_{i\in\Lambda}\right) = \int_{\boxtimes_{j\in\Delta\setminus\Lambda}[\underline{x}_{\rho,j},\overline{x}_{\rho,j}]} f\left((x_i)_{i\in\Delta}\right) d\left(x_j\right)_{j\in\Delta\setminus\Lambda}, $$

$$ \forall (x_i)_{i\in\Lambda} \in \boldsymbol{x}_\rho^{\Lambda} = \boxtimes_{i\in\Lambda}[\underline{x}_{\rho,i}, \overline{x}_{\rho,i}] \quad (3.13) $$

The RP structure allows an RMRP ${}^{\square}f$ with root box $\boldsymbol{x}_\rho \in \mathbb{IR}^d$ to be marginalised very efficiently. The boxes in the paving are 'collapsed' to remove the unwanted dimensions and the effect of integrating out these unwanted dimensions is achieved by then rescaling each $f_{\rho\mathsf{v}}$ to compensate for the lost box volume. Algorithm 3.10 gives the operation `Marginalise`.

Algorithm 3.10: `Marginalise`$(\rho\mathsf{v}, \Lambda)$

input : $\rho\mathsf{v}$, a node in an RMRP ${}^{\square}f$ with box $\boldsymbol{x}_{\rho\mathsf{v}} = \boxtimes_{j\in\Delta}[\underline{x}_{\rho\mathsf{v},j}, \overline{x}_{\rho\mathsf{v},j}] \in \mathbb{IR}^d$, $\Lambda \subset \Delta$ and $\Lambda \neq \emptyset$

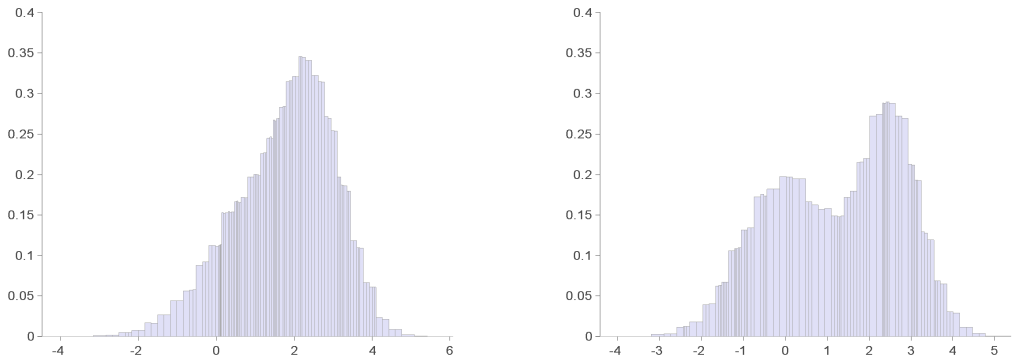
output : Change $\rho\mathsf{v}$ into a node of RMRP ${}^{\square}f^{\Lambda}$ of (3.13)

```
if !IsLeaf(ρv) then
    Marginalise(ρvL, Λ)
    Marginalise(ρvR, Λ)
end
if !IsLeaf(ρv) & ι ∉ Λ then
    ρv ← MRPOperate(ρvL, ρvR, +)
end
else
    x^Λ_ρv ← ⊠_{j∈Λ} [x_{ρv,j}, x̄_{ρv,j}]
    f_ρv ← f^Λ_ρv = f_ρv vol(x_ρv) / vol(x^Λ_ρv)
    x_ρv ← x^Λ_ρv
end
```

`Marginalise` (Algorithm 3.10) can be used to obtain the marginal function of an RMRP ${}^{\square}f$ as another RMRP. If ${}^{\square}f$ is non-negative and integrates to 1 (i.e., is a density function) then a marginal function obtained using `Marginalise` will also be a density function integrating to 1.

If ${}^{\square}f$ is non-negative but does not integrate to 1 (for example, say, is an unnormalised density function) then a marginal obtained using `Marginalise` can be normalised using Algorithm 3.9.

Figure 3.9 illustrates marginalisation using the RMRP density estimate of example Density I (see Appendix B) shown in Figure 3.8.



(a) Marginal on coordinate 1, ${}^{\square}f^{\{1\}}(x_1)$.

(b) Marginal on coordinate 2, ${}^{\square}f^{\{2\}}(x_2)$.

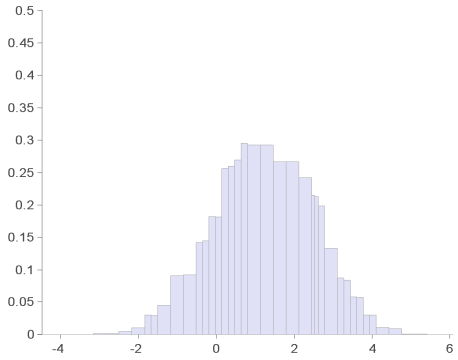
Figure 3.9: Marginals of an RMRP density estimate of Density I, $d = 2$.

Note that the partition of a marginal RMRP density estimate computed using `Marginalise` will reflect the addition process used in Algorithm 3.10. Adding two RMRP nodes gives an RMRP node with a box whose partition is the union of the partitions of the boxes of two operand nodes (see Algorithm 3.6, Section 3.5.2). The complexity of the partition of the marginal RMRP will reflect the extent to which the partitioning of the joint RMRP varies along the coordinates removed (marginalised out) during the operation.

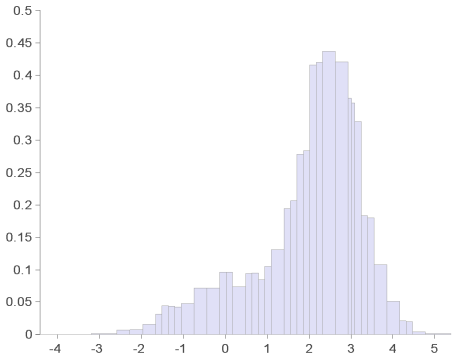
3.6.2.6 Conditional functions

The `MRPSlice` operation (see Algorithm 3.7, Section 3.5.2) can be used to obtain the unnormalised conditional function of an RMRP ${}^{\square}f$. If ${}^{\square}f$ is non-negative and integrates to 1 (i.e., is a density function), `MRPSlice` followed by `Normalise` (Algorithm 3.9) can be used to obtain an RMRP conditional density from ${}^{\square}f$.

Conditional density estimates using slicing are illustrated in Figure 3.10 with normalised slices of the RMRP density estimate shown in Figure 3.8. Figure 3.10(a) shows the normalised slice on $x_2 = 1.0$. This RMRP ${}^{\square}f^{|x_2=1.0}(x_1)$ is an estimate of $f_I(x_1 \,|\, x_2 = 1.0)$ the univariate conditional density of x_1 given $x_2 = 1.0$. Figure 3.10(b) shows the normalised slice on $x_1 = 2.5$. This RMRP ${}^{\square}f^{|x_1=2.5}(x_2)$ is an estimate of $f_I(x_2 \,|\, x_1 = 2.5)$ the univariate conditional density of x_2 given $x_1 = 2.5$.



(a) Conditional density ${}^{\square}f(x_1 \,|\, x_2 = 1.0)$.



(b) Conditional density ${}^{\square}f(x_2 \,|\, x_1 = 2.5)$.

Figure 3.10: Conditional densities from an RMRP density estimate of Density I, $d = 2$.

3.6.2.7 L_1 distance

The L_1 distance between two densities f and g is $\int |f - g|$ (see Section 2.2). The L_1 distance between two RMRPs ${}^{\square}f$ and ${}^{\square}g$ can be calculated using the RMRP operations already defined. The difference ${}^{\square}f - {}^{\square}g$ (Equation (3.10a)) is an RMRP; the absolute value operation $|.|$ is a transformation (Equation (3.10b)) and so $|{}^{\square}f - {}^{\square}g|$ is also an RMRP and the result can be integrated (Section 3.6.2.1) to get

$$L_1\left({}^{\square}f, {}^{\square}g\right) := \int |{}^{\square}f - {}^{\square}g| \ . \tag{3.14}$$

If the 'true' density can be represented as an RMRP ${}^{\square}f$, and an RMRP density estimate ${}^{\square}\hat{f}$ has been made (and both have the same root box), the IAE between ${}^{\square}f$ and ${}^{\square}\hat{f}$ can be calculated as $L_1\left({}^{\square}f, {}^{\square}\hat{f}\right)$.

3.6.2.8 Multiplying

Two RMRPs represented by ${}^{\square}f^{(1)}$ and ${}^{\square}f^{(2)}$ with the same root box $\boldsymbol{x}_\rho$ can be multiplied together to obtain ${}^{\square}f = {}^{\square}f^{(1)} \times {}^{\square}f^{(2)}$ (Equation (3.10a)). ${}^{\square}f$ is an RMRP, so it can be multiplied by another RMRP ${}^{\square}f^{(3)}$. Hence any finite number of RMRPs can be multiplied together to give another RMRP, provided that all the multiplicand RMRPs have the same root box.

3.6.2.9 Dividing

Similarly, given RMRPs represented by ${}^{\square}f$ and ${}^{\square}g^{(1)}$ with the same root box $\boldsymbol{x}_\rho$, Equation (3.10a) can be used to obtain the RMRP ${}^{\square}f / {}^{\square}g^{(1)}$ provided that $f_{\rho\mathsf{v}} \neq 0$ for all $\rho\mathsf{v} \in \mathbb{V}({}^{\square}g^{(1)})$ (to avoid divisions by zero). The resulting RMRP can be divided by another RMRP

${}^{\square}g^{(2)}$ subject to the same conditions. Hence an RMRP ${}^{\square}f$ can be any divided by a finite number of RMRPs to give another RMRP provided that ${}^{\square}f$ and all the divisors have the same root box and none of the divisors has a zero value mapped to any node.

3.6.2.10 Highest coverage regions

In statistical interpretations of probability density functions it can be useful to be able to find the highest coverage region of the density. The α-highest coverage region of an RMRP ${}^{\square}f$ is given by the smallest possible subset of leaf nodes such that they constitute the highest image values and integrate at least to a specified fraction α of the total integral of ${}^{\square}f$ over its root box. Algorithm 3.11 sets out operation `CoverageRegion` to find the α-coverage region of a positive RMRP.

Algorithm 3.11: `CoverageRegion`$({}^{\square}f, \alpha)$

input : A positive RMRP ${}^{\square}f$ with root node ρ, and $\alpha \in [0, 1]$

output : $\mathcal{C}({}^{\square}f)$, smallest subset of leaf nodes of ${}^{\square}f$ that contain the highest α region, i.e., $\int_{\rho\mathsf{v}\in\mathcal{C}({}^{\square}f)} f_{\rho\mathsf{v}}\mathrm{vol}\,(\boldsymbol{x}_{\rho\mathsf{v}}) \geq \alpha$ and $\min_{\rho\mathsf{v}\in\mathcal{C}({}^{\square}f)} f_{\rho\mathsf{v}} > \max_{\rho\mathsf{v}\in\mathbb{L}({}^{\square}f)\setminus\mathcal{C}({}^{\square}f)} f_{\rho\mathsf{v}}$.

initialize: $\mathcal{C}({}^{\square}f) \leftarrow \emptyset$; $a \leftarrow 0$; $i \leftarrow 1$, list $\mathbb{L}^{\downarrow} \leftarrow \mathbb{L}({}^{\square}f)$, $m \leftarrow |\mathbb{L}({}^{\square}f)|$
`// next sort` $\mathbb{L}^{\downarrow}$ `leaves by height`
$\mathbb{L}^{\downarrow}({}^{\square}f) = [\rho\mathsf{v}_1, \rho\mathsf{v}_2, \ldots, \rho\mathsf{v}_n]$, $f_{\rho\mathsf{v}_1} \geq f_{\rho\mathsf{v}_2} \geq \ldots \geq f_{\rho\mathsf{v}_m}$
$N_f \leftarrow$ `Integral`$(\rho, \boldsymbol{x}_\rho)$ `// normalising constant`

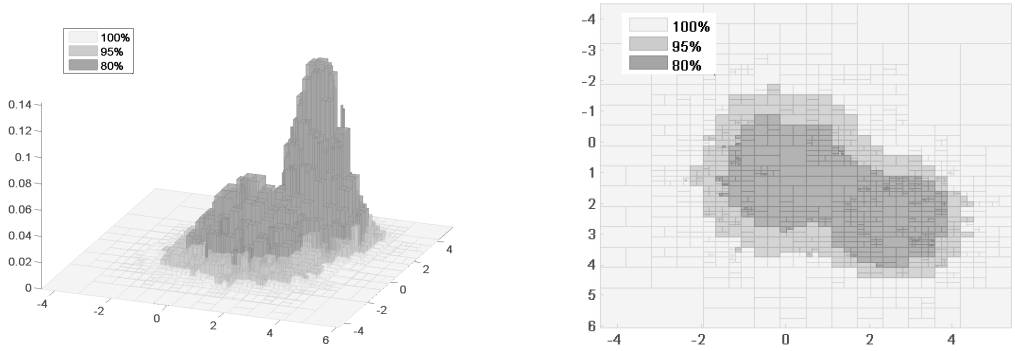
while $i \leq m$ & $a < \alpha$ **do**
  $a \leftarrow a + (f_{\rho\mathsf{v}_i}\mathrm{vol}\,(\boldsymbol{x}_{\rho\mathsf{v}_i}))/N_f$
  $i \leftarrow i + 1$
  $\mathcal{C}({}^{\square}f)$`.append`$(\boldsymbol{x}_{\rho\mathsf{v}_i})$ `// insert this node into the set` $\mathcal{C}({}^{\square}f)$
end
return $\mathcal{C}({}^{\square}f)$

The `CoverageRegion` operation can be used to obtain a highest posterior density region from an RMRP representing a Bayesian posterior density.

Figure 3.11 illustrates coverage regions using the RMRP estimate of Density I shown in Figure 3.8. Figure 3.11(a) shows the 95% (dark-gray) and 80%(mid-gray) coverage regions against the 100% coverage (light-gray) whole from the same viewpoint as Figure 3.8. Figure 3.11(b) gives a plan view of the same structure, showing how the different coverage regions relate to the partitioning of the root box of the RMRP.

3.6.2.11 Data simulation

A non-negative RMRP ${}^{\square}f$ can be used to simulate data. ${}^{\square}f$ is treated like a mixture of uniform densities defined on the boxes associated with each leaf node with weights equal to the integrals of ${}^{\square}f$ over these leaf node boxes.



(a) 80%, 95%, and 100% coverage.

(b) Plan view.

Figure 3.11: Coverage regions for an RMRP density estimate of Density I, $d = 2$.

Algorithm 3.12: `SimulateData`($^{\square}f$)

input : A non-negative RMRP $^{\square}f$ with root node ρ
output : A point $x \in \mathbb{R}^d$ drawn from $^{\square}f$.

initialize: $i \leftarrow 0$, $m \leftarrow |\mathbb{L}(^{\square}f)|$
$[\rho\mathsf{v}_1, \rho\mathsf{v}_2, \ldots, \rho\mathsf{v}_m] = \mathbb{L}(^{\square}f)$ `// sequence of leaf nodes`
$a \leftarrow 0$
$u \leftarrow$ uniform random number in $[0, 1)$
$N_f \leftarrow$ `Integral`$(\rho, \boldsymbol{x}_\rho)$ `// normalising constant`

repeat
  $i \leftarrow i + 1$
  $a \leftarrow a + (f_{\rho\mathsf{v}_i} \mathrm{vol}\,(\boldsymbol{x}_{\rho\mathsf{v}_i}))/N_f$
until $a > u$
$x \leftarrow$ uniform random vector in $\boldsymbol{x}_{\rho\mathsf{v}_i}$
return x

3.6.2.12 Combining operations on piecewise-constant functions

The class $^{\square}\mathcal{F}$ of RMRPs with a given root box is closed under the arithmetical operations encompassed by Equations (3.10a) to (3.10c) and under the additional operations represented by `Normalise`, `MRPSlice` (conditioning), and `Marginalise`. These operations can be repeated or combined in any order provided only that the root boxes of the operand RMRPs are the same. Integrals, coverage regions, or the pointwise image of a point in the root box, can be calculated at any stage during such series of operations, or the L_1 distance between the result and some other RMRP can be found. As a simple example, `Marginalise` and `CoverageRegion` can be combined to create 1-d or 2-d visualisations of higher-dimensional densities represented using RMRPs. Chapter 9 includes an extensive example applying almost all the RMRP operations described above.

3.7 Statistical regular paving histograms and piecewise-constant function estimates

The SRP structure described in Section 3.4.1 is designed for analysing and organising data. Information on the leaf boxes of the paving and the data count statistics can be used to create a histogram as described in Section 3.4.3. This section describes how an SRP can be used to obtain a PCF structured as an RMRP that is equivalent to the histogram of the sample data on the SRP partition. This allows all the RMRP operations described above to be carried out on the RMRP histogram density estimate formed from the SRP.

Given an SRP ${}^{\circ}s$ with root node ${}^{\circ}\rho$ and root paving $\boldsymbol{x}_{{}^{\circ}\rho}$ whose leaf nodes are associated with a total of n data points ($n = \#\boldsymbol{x}_{{}^{\circ}\rho} = \sum_{{}^{\circ}\rho\mathsf{v}\in\mathbb{L}({}^{\circ}s)} \#\boldsymbol{x}_{{}^{\circ}\rho\mathsf{v}}$), `SRPtoPCFDensity`$({}^{\circ}\rho, n)$ (Algorithm 3.13) returns the root node ρ of an RMRP ${}^{\square}f$ with $\boldsymbol{x}_{\mathbb{V}({}^{\square}f)} = \boldsymbol{x}_{\mathbb{V}({}^{\circ}s)}$ (i.e., the same partition of the same root box as the SRP) such that ${}^{\square}f$ represents a probability density function (${}^{\square}f$ is non-negative and $\int_{\boldsymbol{x}_\rho} {}^{\square}f = 1$) and for equivalent nodes $\rho\mathsf{v} \in \mathbb{V}({}^{\square}f)$ and ${}^{\circ}\rho\mathsf{v} \in \mathbb{V}({}^{\circ}s)$, $\boldsymbol{f}_{\rho\mathsf{v}} = \frac{\#\boldsymbol{x}_{{}^{\circ}\rho\mathsf{v}}}{n\mathrm{vol}\,(\boldsymbol{x}_{{}^{\circ}\rho\mathsf{v}})}$.

Algorithm 3.13: `SRPtoPCFDensity`$(\rho\mathsf{v}, n)$

input : ${}^{\circ}\rho\mathsf{v}$, a node in an SRP ${}^{\circ}s$, $n = \sum_{{}^{\circ}\rho\mathsf{v}\in\mathbb{L}({}^{\circ}s)} \#\boldsymbol{x}_{{}^{\circ}\rho\mathsf{v}}$
output : $\rho\mathsf{v}$, a node in an RMRP ${}^{\square}f$

Make a new node $\rho\mathsf{v}$
$\boldsymbol{x}_{\rho\mathsf{v}} \leftarrow \boldsymbol{x}_{{}^{\circ}\rho\mathsf{v}}$
$f_{\rho\mathsf{v}} \leftarrow \frac{\#\boldsymbol{x}_{{}^{\circ}\rho\mathsf{v}}}{n\mathrm{vol}\,(\boldsymbol{x}_{{}^{\circ}\rho\mathsf{v}})}$
if ! `IsLeaf`$({}^{\circ}\rho\mathsf{v})$ **then**
  Graft onto $\rho\mathsf{v}$ as left child the node `SRPtoPCFDensity`$({}^{\circ}\rho\mathsf{vL}, n)$
  Graft onto $\rho\mathsf{v}$ as right child the node `SRPtoPCFDensity`$({}^{\circ}\rho\mathsf{vR}, n)$
end
return $\rho\mathsf{v}$

The operation `SRPtoPCFDensity` can be used to obtain a histogram density estimate, in the form of an RMRP, from an SRP. Like Equation (3.3), `SRPtoPCFDensity` does not address the issue of how to find an appropriate partition of the root box $\boldsymbol{x}$ for the SRP.

3.8 Summary

This chapter has introduced the RP (regular paving) structure and two types of RP: the SRP (statistical regular paving) and the MRP (mapped regular paving). Figure 3.12 shows the relationships between these types in the form of a simplified class diagram based on the computer implementation of these structures used for this thesis. The diagram shows the general form of the $\mathbb{Y}$-mapped regular paving ($\mathbb{Y}$-MRP) for some non-empty set $\mathbb{Y}$. An RMRP is a real-mapped regular paving ($\mathbb{R}$-MRP). The restricted regular paving partitioning makes it

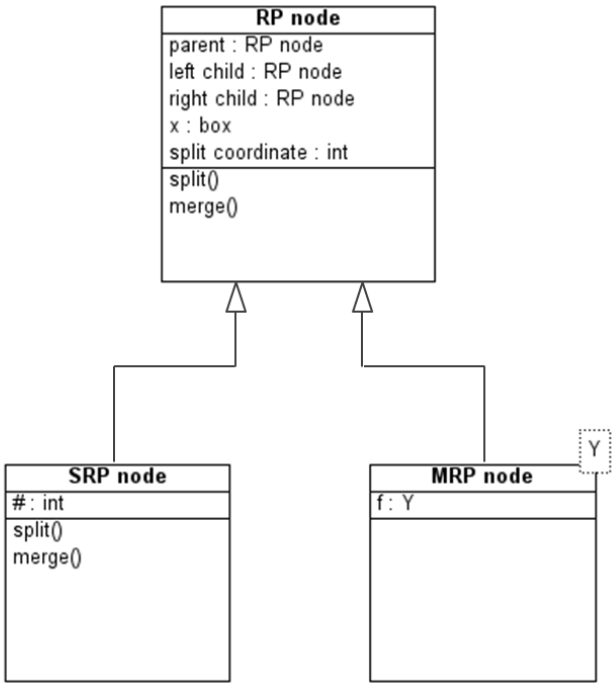


Figure 3.12: Class diagram for RP types.

possible to carry out a wide range of operations on these structures very efficiently. The SRP is a partitioned 'container' and summariser for multivariate data that can be converted into an RMRP representation of the histogram of the SRP data on the SRP partition. The properties of RMRPs allow arithmetic operations defined on $\mathbb{R}$ to then be extended to these RMRP density estimates. The RMRP structure also means that operations such as marginalising or obtaining a conditional density can be carried out very efficiently. The next three chapters discuss how the partitioning of the root box of an SRP can be carried out.

Chapter 4

Data-adaptive partitioning using statistical regular pavings

4.1 Introduction

The research on density estimation using histograms reviewed in Section 2.2 has shown that, in theory, data-adaptive histograms can provide better density estimates than non-adaptive (fixed) partitioning schemes, especially for multivariate data, but that it is not straightforward to find a good data-adaptive partitioning method.

This chapter discusses data-adaptive partitioning of multivariate data using the SRP structure described in Section 3.4. Once partitioned, an SRP can be used to obtain a PCF represented as an RMRP that is equivalent to the histogram of the sample data on that partition using the operation `SRPtoPCFDensity` (Algorithm 3.13). The key advantage of the SRP-based RMRP histogram is the arithmetical operations that can be carried out with RMRPs allow individual histograms to be combined and manipulated to form the final density estimate. It is the methods by which data-adaptive partitioning may be achieved, rather than the *individual* histograms formed, that are the focus of interest in this chapter.

Data-adaptive partitioning using SRPs is defined in Section 4.2. The general restrictions that will apply to data-adaptive partitioning of SRPs are reviewed in Section 4.3. The implications of relationships between data coordinates and the scale of the data are discussed in Section 4.4. An overview of some of the issues encountered in trying to design data-dependent partitioning using SRPs, illustrated using two possible partitioning methods for a simple univariate target density, is given in Section 4.5. The two methods covered in Section 4.5, a priority queue algorithm seeking the immediate next best refinement to the partition and a Markov chain Monte Carlo (MCMC) algorithm, are discussed in detail in Chapters 5 and 6, respectively.

4.2 Definition

Section 3.7 showed how an SRP with a given partition can be converted into a histogram in the form of an RMRP. Forming a data-adaptive histogram using an SRP means using a data-adaptive *partitioning strategy*. A partitioning strategy for an SRP $^\circ s$ is a process for determining which (if any) of the nodes in the current set of leaf nodes $\mathbb{L}(^\circ s)$ should be the next to be split and when the partitioning should stop.

4.3 Restrictions on data-adaptive partitioning methods

4.3.1 Regular bisections

An important point to note before discussing data-adaptive partitioning of SRPs in any detail is that the partitioning strategy must be compatible with the regular bisection rule given in Section 3.3.1: if a node in an SRP is split then its box is bisected exactly at the midpoint of its first coordinate of maximum width. This restriction has many disadvantages but also makes possible the very wide range of arithmetic operations described in Sections 3.5 and 3.6.

4.3.2 Splittable nodes

There are also some restrictions on nodes that can be considered to be *splittable* in the context of a data-adaptive partitioning strategy for SRPs.

Clearly a data-adaptive partitioning strategy should only split nodes with at least some data associated with them (non-empty nodes). This restriction applies to SRPs in particular. A computer implementation will also place restrictions on RP partitioning algorithms in general. Computer-representable floating point numbers are only a limited subset of the real numbers $\mathbb{R}$. Because of this, not all bisections of a box represented in a computer will result in the two child boxes described in Section 3.3.1. A node with a box that cannot be bisected into two properly representable child boxes must be considered to be an unsplittable node. Similarly, bisection of a box may result in a box with volume smaller than the smallest floating point number that can be properly represented by a computer. Calculations involving the volumes of the boxes in an RP form part of many of the algorithms described in Chapter 3 including operation `SRPtoPCFDensity` to convert of an SRP to an RMRP (Algorithm 3.13). Nodes with boxes that, if bisected, would yield child-boxes that have volume smaller than the smallest normalised representable floating point number must therefore also be considered to be unsplittable. Appendix C discusses the restrictions that a computer implementation will place on RP partitioning in more detail.

Further criteria could be added to the set of conditions that must be met for a node to be considered to be splittable. For example, it might be desirable to try to ensure that every leaf node $\rho\mathsf{v}$ in an SRP $^\circ s$ has at least $\underline{\#}$ data points associated with it, if it has any data associated with it at all, i.e., $\#\boldsymbol{x}_{\rho\mathsf{v}} \geq \underline{\#}$ or $\#\boldsymbol{x}_{\rho\mathsf{v}} = 0$ for all $\rho\mathsf{v} \in \mathbb{L}(^\circ s)$.

The set of splittable nodes of an SRP $^\circ s$, however defined, is denoted by $\mathbb{L}^\triangledown(^\circ s)$.

4.3.3 Feasible partitioning methods

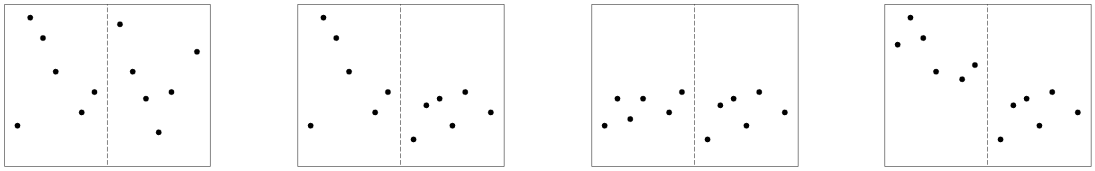
A further important aspect is the size of the space of SRPs. The number of distinct binary trees with k splits is equal to the Catalan number C_k (Equation (3.1)). An SRP with $m = |\mathbb{L}(^\circ s)|$ leaves has a tree with $k = m - 1$ splits, and the number of leaves in the tree is equivalent

to the number of elements in the partition of the root box. The Catalan numbers grow very rapidly: as $k \to \infty$, $\frac{C_{k+1}}{C_k} = \frac{2(2k+1)}{(k+2)} \to 4$ from below (Knuth 2006a). $C_{k=37}$ is larger than the largest integer that can be represented in a computer using standard number formats.[1] The vast size of the space of SRPs means that attempting to compare histograms based on all the possible partitions of a root box, even if some very low maximum number of leaves $\overline{m}$ is specified, is clearly impractical. Brute-force searches (exhaustive enumeration and comparison of final outcomes), such as those used with dyadic histograms (Klemelä 2009, chap. 18), are not considered in this thesis.

The next important observation is that a node in an SRP is for all practical purposes myopic. The only information available for the current set of leaf nodes consists of:

- The recursively-computable statistics held by the node (this thesis assumes only that the data count $\#\boldsymbol{x}_{\rho\mathsf{v}}$ is held for each node $\rho\mathsf{v} \in \mathbb{V}(\,^\circ s)$ in an SRP $\,^\circ s$); and
- The box $\boldsymbol{x}_{\rho\mathsf{v}}$ associated with each node.

Nodes do not 'know' about the *distribution* of the data associated with them. In Figure 4.1, for example, a node $\rho\mathsf{v}$ associated with any of the complete boxes shown in Figures 4.1(a), 4.1(b), 4.1(c), 4.1(d) will have the same size of box and the same data count $\#\boldsymbol{x}_{\rho\mathsf{v}} = 12$. Even if it is computationally practical for a partitioning strategy to compare leaf nodes by looking one step ahead and considering the immediate effect of splitting each one of them, this gives limited additional information. The restrictive bisection rule means that the coordinate on which the box associated with a node would be bisected is not determined by the data and splitting a node will not necessarily immediately reveal the most interesting features of the data associated with it. In each of Figures 4.1(a), 4.1(b), 4.1(c), 4.1(d) the new child boxes in the event of a bisection are delimited with a dotted line and each new child would have 6 data points associated with it. This information can only be found by the computationally costly process of checking which of the prospective child boxes each of the data points associated with the node would fall into.



(a) (b) (c) (d)

Figure 4.1: Node boxes and different distributions of sample data.

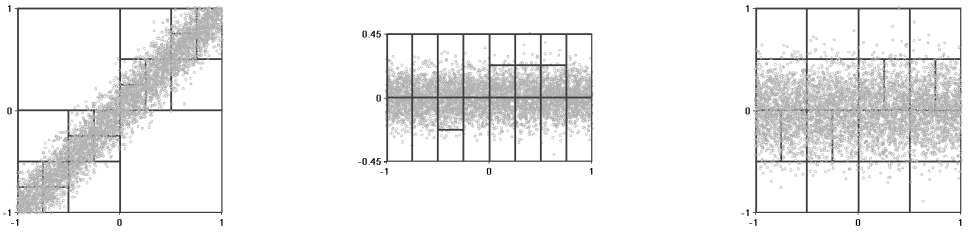
In this simple example comparing the nodes by one more step ahead (on the basis of the effect of two splits in total) would give more useful information, but in practice it might be

[1] $C_{k=36} = 11,959,798,385,860,453,492$

necessary to look many more steps ahead, while the total number of calculations required increases exponentially with each additional future split considered. Experimentation with a range of specific densities during the course of the research for this thesis suggests that not only is it is very computationally expensive to try such 'look-ahead' strategies, but it is often ineffectual because the most important asymmetries and variations in the distribution of the data remain hidden for more future splits than it is feasible to consider.

4.4 Data considerations

The regular bisection rule given in Section 3.3.1 also has implications for the data best suited to an SRP. Correlations within the data and differences in the scale of the data between coordinates can both affect SRP partitioning.



(a) Correlated data. (b) Uncorrelated data. (c) Standardised data.

Figure 4.2: Regular bisections and correlated or un-standardised data.

4.4.1 Relationships between coordinates within the data

The root box and the sub-boxes in a regular paving are always axis-aligned hyper-rectangles. This is not well-suited to sample data with strong correlations between data coordinates. Figure 4.2(a) shows a partition of the root box of an SRP associated with a sample of bivariate data with support $[-1, 1]^2$ where the first and second coordinates of the data are strongly correlated. The sample data is shown superimposed on the partition. The axis-aligned root box necessarily includes large regions with no sample data. Figure 4.2(b) shows a partition of the root box of an SRP associated with a sample of bivariate data where there is no correlation between the data coordinates. Again the data is superimposed on the partition.

Where there are strong correlations between the data coordinates the data can be transformed using rotations or reflections (transformations that preserve angles, distance and area) to give a better axis alignment before associating it with an SRP, partitioning, and creating an RMRP from the SRP. This would avoid some of the inefficiencies associated with partitioning mainly-empty space but such a transformation would also severely restrict the operations that could be applied to the resulting RMRP.

Under a rotation or reflection each coordinate of the transformed data is a function of all the coordinates of the untransformed data. For example, a rotation of $-\frac{\pi}{4}$ (45° clockwise) in $\mathbb{R}^2$, as might be used to align the data shown in Figure 4.2(a) with the first coordinate axis, is achieved by transforming each $x = (x_1, x_2)$ into $x' = (\frac{1}{\sqrt{2}}x_1 + \frac{1}{\sqrt{2}}x_2, -\frac{1}{\sqrt{2}}x_1 + \frac{1}{\sqrt{2}}x_2)$. This means that it is not possible, in the transformed space, to then create a marginal density function on a subset of the coordinates in the untransformed space (e.g., marginalise on x_1 or x_2), or to create a conditional density function by conditioning on values of coordinates in the untransformed space. A back-transformation of the RMRP formed using transformed data into a PCF in the original coordinate space is straightforward but will only yield a structure with an RP partition in the case of a very limited range of transformations. Without the RP structure, the RP operations described in Chapter 3 cannot be applied to the back-transformed PCF.

4.4.2 Data scale

If a node is split, the split takes place on the first widest coordinate of the box associated with that node. This means that the scaling of the data on each coordinate will influence the coordinate on which the next bisection takes place. In particular, if the scale of the data on one dimension is much larger than on any other dimension, more partitioning will take place on that dimension. Figure 4.2(b) illustrates this. When the differences in scale are larger the effects are very much more obvious.

In this situation the data can be standardised to have a similar scale on each coordinate before associating it with an SRP, partitioning, and creating an RMRP. Figure 4.2(c) shows an SRP partition using the same data as Figure 4.2(b) after that data has been standardised to fit into a $[-1, 1]^2$ box (the standardised data is superimposed on the partition). Assuming that the transformation is a simple rescaling or shift on some of the coordinates it is possible to use the marginalisation and conditioning operations in the transformed coordinate space to give other RMRPs in that same transformed coordinate space. Back-transformation of any of these RMRP to give a PCF in the original coordinate space is again straightforward, but — as with back-transformation of an RMRP formed from rotated or reflected data — will typically not result in an RP partitioned structure and therefore operations defined for RMRPs will not be generally applicable to the PCF obtained after the back-transformation.

4.5 A simple example

This section explores a very simple example to illustrate some of the issues discussed above in the context of two important data-adaptive partitioning strategies.

4.5.1 Randomised priority queue partitioning

Density estimation methods using statistically equivalent block (SEB)s were briefly described in Section 2.2. Teng (2013) developed an SEB-based SRP partitioning scheme driven by a randomised priority queue (RPQ) aiming for a final SRP where each leaf node has at most $\overline{\#}$ of the sample data points associated with it and showed that the resulting SRP histogram estimate is asymptotically consistent in the L_1-setting if $\overline{\#}$ and the maximum number of leaves $\overline{m}$ in the SRP meet certain conditions. RPQs are discussed in detail in Chapter 5.

Like the other greedy algorithms reviewed in Section 2.2, the SEB-based RPQ algorithm is not guaranteed to produce an optimal partition.

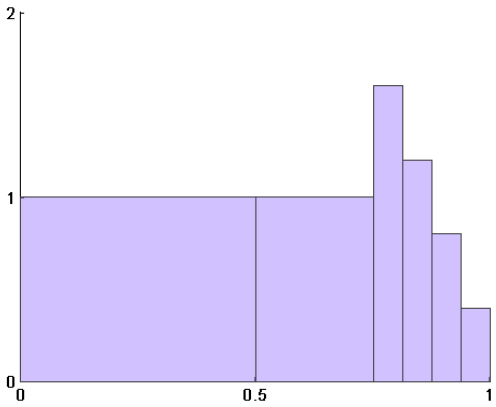


Figure 4.3: True density for a simple example.

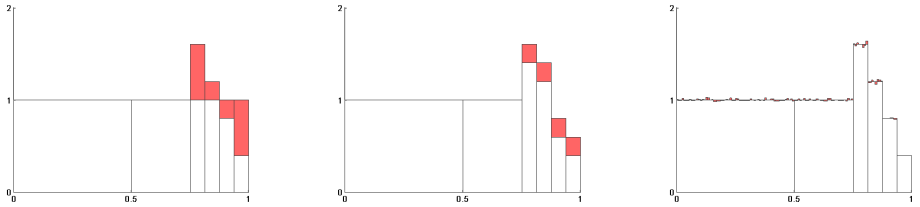
The simple example explored in this section uses a true density that is a mixture of uniform densities: 75% Uniform(0, 0.75), 10% Uniform(0.75, 0.8125), 7.5% Uniform(0.8125, 0.875), 5% Uniform(0.875, 0.9375), and 2.5% Uniform(0.9375, 1). An RMRP representation of this density is shown in Figure 4.3.

A sample of 10,000 data points drawn from this density was associated with an SRP $\circ s$ with root box [0, 1]. An SEB-based RPQ process of the type described by Teng (2013) (see also Appendix D) was used to partition this root box. The starting state and the states resulting from subsequent splits were converted into RMRPs using the `SRPtoPCFDensity` operation (Algorithm 3.13). Figure 4.4 illustrates the results by showing the 20 first states (including the starting state) in the sequence of states obtained from the RPQ.

The SEB-based RPQ algorithm will choose the next node to split at random from the nodes with most data associated with them. As Figure 4.4 shows, this can result in much unnecessary splitting in regions where the data is relatively dense but is also approximately uniform. Not only is the additional splitting computationally inefficient, but it will also tend to give an undersmoothed result. Figure 4.5 illustrates this by shading the areas that together make up the total L_1 error against the true density for three selected states from Figure 4.4.

Figure 4.4: First 20 RPQ states.

For a relatively small tree, such as the one considered here, different replications of the RPQ using a different series of pseudo-random numbers in the RPQ algorithm will usually give very similar results. The randomisation only affects which of a small group of equally 'large' nodes is chosen to be split first (in larger examples different replications can sometimes give more differentiated outcomes).



(a) L_1 error in starting state. (b) L_1 error with 8 leaves. (c) L_1 error with 100 leaves.

Figure 4.5: L_1 errors for selected states from the priority queue.

4.5.2 Markov chain Monte Carlo partitioning

The same example data is now used to illustrate partitioning using the MCMC approach described in Teng (2013) and Sainudiin et al. (2013). This MCMC algorithm is discussed in detail in Chapter 6; the purpose of this simple example is again to illustrate some the most

important characteristics of this partitioning method. A short chain, starting with the SRP $^\circ s$ with just the root box and no partitioning, was run and at each state in the chain the SRP was converted into an RMRP so as to be able to illustrate the SRP states in the RPQ in this familiar form.

Figure 4.6 shows RMRPs formed using the complete set of unique partitions visited by the time that the chain first reached the partition equivalent to the uniform mixture of the true density (Figure 4.3), which occurred in the 36^{th} state visited by this chain. For the first 10 states in the chain, the SRP $^\circ s$ wandered back and forth between a state comprising solely the root node and a state with two leaf nodes (two sub-boxes in the partition). It then made a brief exploration of the effect of further splitting on the left hand side of the partition, moved back to the two-state shuffle described above, and finally started deeper splitting of the area in the interval [0.75, 1] in state 31.

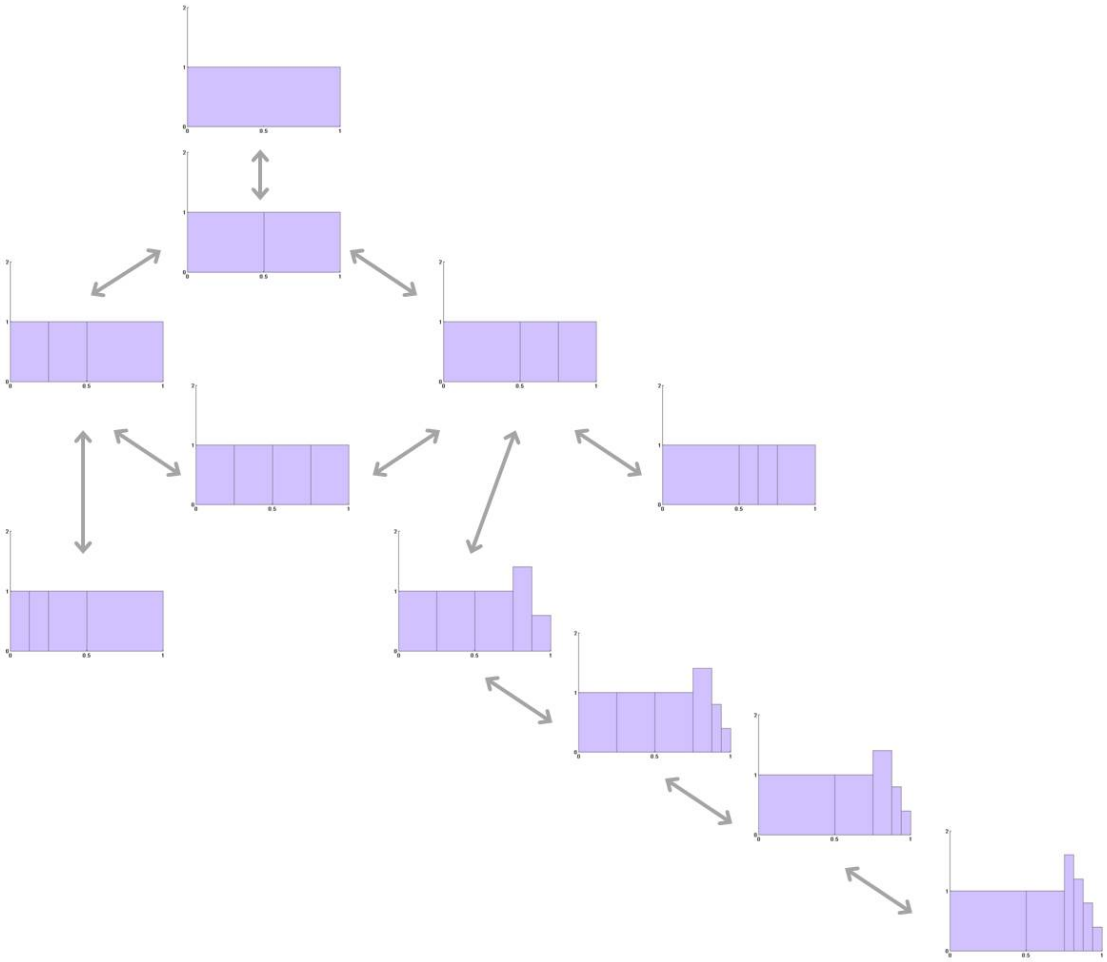


Figure 4.6: Unique states visited in MCMC chain of length 36 (first MCMC replication).

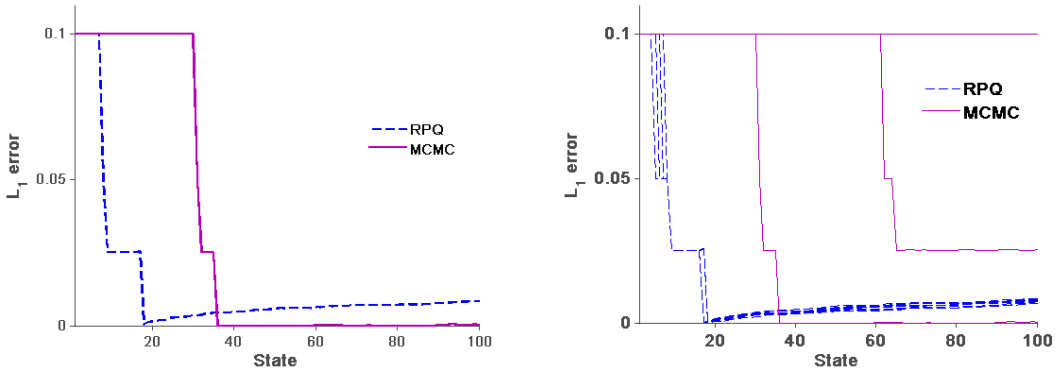
When this example was replicated a further nine times (using different sequence of pseudo-

random numbers for the generation of the sample data and within the MCMC algorithm) there were considerable differences between the chains. One of the replications in fact failed to explore splitting on the right of the partition at all for over 100 states in the chain. Another achieved some splitting on the right only after 60 states, and then only moved within another small group of states. Both of these replications eventually explored deeper splitting on the right side of the root box when the chains were run for longer.

The details of each replication are of course not important; the main point is that many aspects of this behaviour are very typical of the SRP MCMC process described in Teng (2013) and Sainudiin et al. (2013). A chain can easily become temporarily trapped in a limited number of states for long periods.

4.5.3 Comparing SEB-RPQ and MCMC partitioning

Figure 4.7 compares SEB-RPQ and MCMC partitioning using the L_1 errors (integrated absolute error (IAE)s) of the sequence of RMRP histograms obtained from the sequence of SRPs produced by each process against the true density. Figure 4.7(a) shows the errors for successive states generated by the SEB RPQ process and for the first MCMC replication. Figure 4.7(b) shows the variability of outcomes from different replications of both the SEB RPQ and the MCMC processes. Ten SEB RPQ replications are shown, illustrating how similar each one is to the others. The gradually increasing L_1 errors are due to the increase in undersmoothing as splitting continues in the later stages the SEB RPQ process. The paths taken by different Markov chains can be very dissimilar but once a chain has found the partitions most compatible with the distribution of the sample data there is no systematic requirement to continue to split nodes in the SRP.



(a) One replication. (b) Multiple replications.

Figure 4.7: L_1 errors for a single replication and multiple replications.

As Teng (2013) noted, without some knowledge of the true density of the data, it is difficult to find appropriate values for parameters to control the SEB-based RPQ and prevent

undersmoothing in flat higher density areas. However, an SEB-based RPQ run for a short time can be an effective way to obtain an oversmoothed histogram that reflects the major features of the density of the sample data.

The MCMC process illustrated here mixes very slowly so that the chain may need to be run for a long time before it starts to explore states close to the true underling density. A chain can easily get trapped in a sub-space of the total state space for long periods but the SRP can mix split and merge transitions, or retain its current states, rather than being required, as the SEB-based RPQ is, to keep splitting remorselessly until some stopping criteria is met.

4.6 Summary

This chapter has discussed data-adaptive partitioning using SRPs and introduced two possible data-adaptive partitioning methods: an SEB-based RPQ and an MCMC process. The RP structure restricts the partition that can be created and SRPs are affected, like the tree-structured estimators discussed in Chapter 2, by the 'greedy-choice' issue: locally optimal choices are not necessarily globally optimal and exhaustive comparison of all possible outcomes is impractical.

The small and highly artificial example in Section 4.5 illustrates these general issues and also highlights the main differences between the SEB-based RPQ and the MCMC process as data-partitioning strategies. Both have disadvantages, but both also have some strengths. RPQ and MCMC methods are now considered in more detail in Chapters 5 and 6 respectively. A method for forming an RMRP density estimate that does not use an SRP for data-adaptive partitioning, approximation of a kernel density estimate (KDE) using an RMRP, is discussed in Chapter 7.

Chapter 5

Randomised priority queues and statistical regular pavings

5.1 Introduction

Chapter 4 introduced the randomised priority queue (RPQ) as a partitioning strategy for SRPs in the context of a simple example and a statistically equivalent block (SEB)-based queue. This chapter considers the potential for the use of RPQs for data-adaptive partitioning of SRPs in more detail.

5.2 Randomised priority queues for data-adaptive partitioning

A general RPQ SRP partitioning algorithm is given in Appendix D. An RPQ partitioning method orders the splittable leaf nodes of an SRP according to some priority function $\psi : \mathbb{L}^{\triangledown}({}^{\circ}s) \to \mathbb{R}$ and selects the next node to be split from $\operatorname{argmax}_{\rho \mathsf{v} \in \mathbb{L}^{\triangledown}({}^{\circ}s)} \psi(\rho \mathsf{v})$, the set of splittable leaf nodes of ${}^{\circ}s$ which are equally 'large' when measured using ψ. If there is more than one such 'largest' node the choice is made uniformly at random from this set; this is the 'randomised' aspect of the process. Two criteria can be specified to stop the RPQ partitioning. A straightforward stopping condition is to stop partitioning when the number of leaves in the SRP reaches a specified maximum $\overline{m}$. The other stopping condition relates to the priority function so that partitioning stops when the value of the largest node under the priority function ψ is less than or equal to a specified value $\overline{\psi}$. An RPQ will also stop partitioning if there are no splittable leaf nodes in the SRP.

The RPQ process generates a sequence of states $\{S(t)\}_{t \in \mathbb{Z}_+}$ on ${}^{\circ}\mathbb{S}_{1:\overline{m}-1}$. If the initial state $S(t=0)$ is the root ${}^{\circ}s \in {}^{\circ}\mathbb{S}_0$ then this can be seen as a sequence $\{S(k)\}_{k \in \mathbb{Z}_+}$ on ${}^{\circ}\mathbb{S}_{0:\overline{m}-1}$ such that $S(k) \in {}^{\circ}\mathbb{S}_k$ (the $(k+1)^{\text{th}}$ state has $k+1$ leaves, k splits).

5.3 Statistically equivalent block partitioning

The SEB priority function is

$$\psi(\rho \mathsf{v}) = \#\boldsymbol{x}_{\rho \mathsf{v}} \ . \tag{5.1}$$

Teng (2013) developed an SEB-based SRP partitioning scheme driven by an RPQ aiming to create a final SRP where each leaf node has a most $\overline{\#}$ of the sample data points associated with it and the total number of leaves is at most $\overline{m}$, and showed that an RMRP density estimate based on an SRP successfully created using this RPQ partitioning scheme is asymptotically L_1

consistent provided that $\overline{\#}$ and $\overline{m}$ grow with the sample size n at appropriate rates.

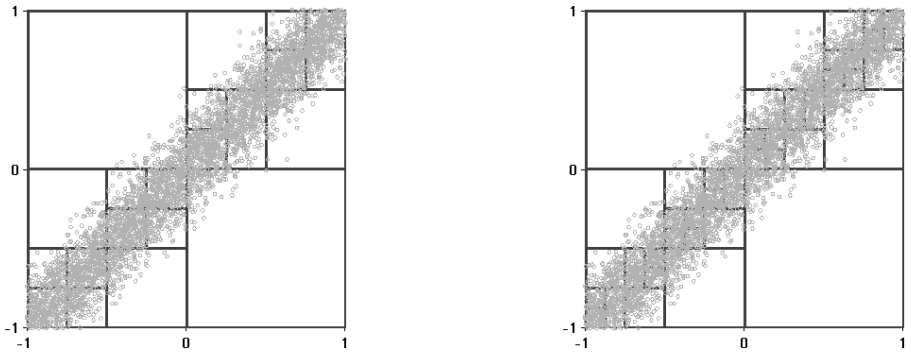
Section 4.5 discussed some of the most important features of SEB-based RPQ partitioning:

- Without some idea of the characteristics of the density to be estimated it is extremely hard to determine suitable values for the parameters controlling the queue.

- SEB-based RPQ partitioning will tend to result in inefficient partitioning and under-smoothing in areas where the data is has relatively high but flat density.

- An SEB-based RPQ run for a limited number of states can be an effective way to create an oversmoothed histogram that reflects the major features of the density of the sample data.

Under an SEB RPQ partitioning strategy the nodes with least data associated with them will remain unsplit for longer (and will possibly never be split). This tends to result in relatively large regions of very low density in the tails of the RMRP PCF formed from the SRP. Figure 5.1 shows two partitions of an SRP associated with the correlated data discussed in Section 4.4.1 (the data is again shown superimposed on the partitions). As the number of leaves in the partition increases from 20 (Figure 5.1(a)) to 40 (Figure 5.1(b)), large sub-boxes containing very little sample data remain unsplit. The effect can be especially distorting to the resulting histogram when the axis-aligned hyper-rectangle root box required by the SRP is a poor fit to the data (when large areas of the root box contain no data points). As discussed in Section 4.4.1, a transformation of the data to mitigate the problem may not be desirable because such a transformation would preclude the subsequent creation of marginal or conditional densities on the coordinates of the untransformed data.



(a) 20 leaves.

(b) 40 leaves.

Figure 5.1: Partition using an SEB RPQ.

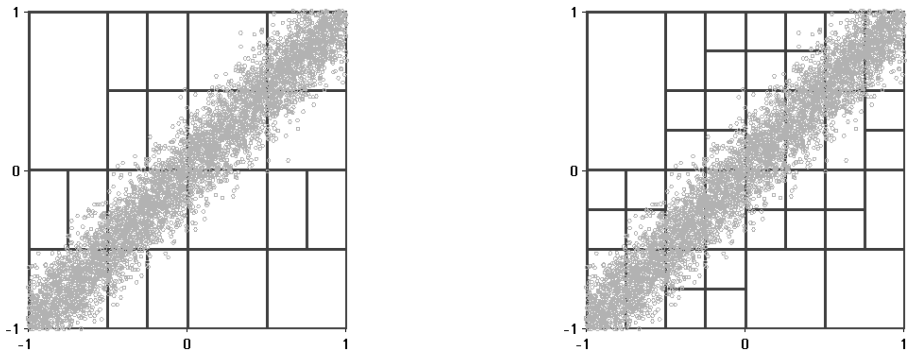
5.4 Partitioning to carve out empty space

Using an RPQ to carve out empty space is an inversion of the SEB-based RPQ: instead of prioritising splitting of nodes with the largest number of data points associated with them, it prioritises splitting of non-empty leaf nodes with large boxes but few data points. A 'carving' priority function for an SRP with a total of $n = \#\boldsymbol{x}_\rho$ data points associated with its root box $\boldsymbol{x}_\rho$ is

$$\psi(\rho\mathsf{v}) = \left(1 - \frac{\#\boldsymbol{x}_{\rho\mathsf{v}}}{n}\right) \mathrm{vol}\,(\rho\mathsf{v}) \ . \tag{5.2}$$

Algorithm D.1 (Appendix D) with priority function ψ as in Equation (5.2) can be used to obtain a 'carved' SRP with $\overline{m}$ leaves from an SRP $^\circ s \in {}^\circ\mathbb{S}_0$ by applying the procedure $\texttt{RPQ}(^\circ s, \psi, \overline{\psi}, \overline{m})$ with specifying $\overline{\psi} = 0.0$ (so that the partitioning process only stops when the SRP has $\overline{\psi}$ leaves or aborts if there a no splittable nodes to continue to split).

The carving effect is illustrated in Figure 5.2. This depicts an SRP with the same correlated data as in Figure 5.2, but the partition is created using an RPQ driven by the carving priority function in Equation (5.2). Figures 5.2(a) and 5.2(b) again show the partition when the SRP has 20 and 40 leaves, respectively. Partitioning is concentrated in the regions of sparse sample data and the effect is to reduce the size of the sub-boxes of the partition into which these sparse data points fall, in effect more tightly enclosing the support of the data.



(a) 20 leaves.

(b) 40 leaves.

Figure 5.2: Partition using a 'carving' RPQ.

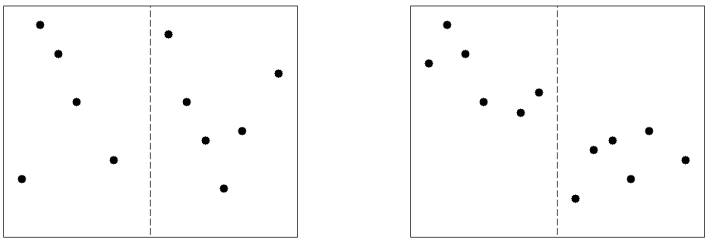
A carving RPQ alone will not give an effective data-driven partitioning strategy, but used in conjunction with an SEB-based RPQ it can improve the SRP histogram. An initial carving RPQ can be run for a short time (specifying $\overline{\psi} = 0.0$ and a relatively low value of $\overline{m}$), followed by an SEB-based RPQ. The empty elements of the partition 'carved out' will be ignored by the SEB-based RPQ, under which partitioning will be concentrated on the areas where most of the sample data has fallen.

5.5 Partitioning with other priority functions

A large variety of other priority functions could be used. In all cases the operation of the RPQ algorithm will be affected by the issues discussed in Section 4.3.3. In particular, under the general RPQ algorithm (see Appendix D) the value of $\psi(\rho\mathsf{v})$ must be calculated for each $\rho\mathsf{v} \in \mathbb{L}^{\triangledown}$. This means that it is computationally expensive to use anything other than immediately available (*local*) information for calculating $\psi(\rho\mathsf{v})$. For example, a maximum-likelihood approach with a priority function calculating the increase in histogram likelihood (Equation (3.4)) that results from splitting a node would have to check, for each of the splittable leaf nodes in the SRP $^{\circ}s$, how many of the data points associated with that leaf node would fall into the boxes associated with the prospective left and right child nodes if that leaf node were to be split.

The same look-ahead to the amount of data that would fall into the boxes associated with the prospective left and right child nodes could be used to give a priority function measuring the L_1 distance between histogram estimates based on the current partition and the partition obtained by splitting each current leaf node (similar to the approach taken for HIRED histograms (Baltrunas et al. 2006)).

The most important limitation of such a look-ahead approach is that (as already discussed in Section 4.3.3), it is not possible to guarantee that a locally optimal decision is also a globally optimal decision: Figure 5.3 illustrates this with a simple variation on Figure 4.1. Splitting the node with associated data and box shown in Figure 5.3(a) will give the largest immediate increase in the likelihood (or largest L_1 distance between current and the prospective histogram estimate) because the data will be unevenly distributed amongst the new child nodes (five data points to the new left child, six to the new right child), whereas splitting the node with associated data and box as shown in Figure 5.3(b) will give no immediate change in the likelihood (and the L_1 distance between current and the prospective histogram estimate is 0) because exactly half of the data would go to each prospective child. Clearly, however, splitting Figure 5.3(b) is a better longer term move because it allows further bisections that expose much more of the variability of the data.



(a) Immediately optimal bisection. (b) Better longer term bisection.

Figure 5.3: Comparison of node data distributions for look-ahead partitioning.

5.6 Summary

This chapter has discussed the RPQ algorithm as a data-adaptive partitioning strategy for SRP rather than as a density estimation method. A drawback of the SRP RPQ algorithm as a density estimator is that, without some idea of the characteristics of the density to be estimated, it is extremely hard to determine suitable values for the parameters controlling the partitioning process. As with the other greedy algorithms discussed in Chapter 2, the locally optimal choices made by an RPQ algorithm may be globally suboptimal. An SEB-based RPQ has nevertheless been shown to be able to produce an asymptotically consistent density estimate. Cross-validation or minimum distance estimation (Devroye and Lugosi 2001, chap. 6), or other smoothing techniques, could potentially be used with SRP RPQs to produce RMRP density estimates. These possibilities are not explored in this thesis.

A major advantage of the SEB-based RPQ algorithm discussed here is that, run for a limited number of states, it can be an effective way to create an oversmoothed histogram that reflects the major features of the density of the sample data. Initiating the process with a short carving RPQ can give further improvements. Between them, the SEB and carving RPQs can result in an SRP that provides a much better starting point for further data-adaptive partitioning than the original, unpartitioned, SRP. The potential to exploit this in conjunction with MCMC partitioning is explored in the next chapter.

Chapter 6

Markov chain Monte Carlo partitioning and statistical regular pavings

6.1 Introduction

Chapter 4 included an example of a short Metropolis-Hastings Markov chain partitioning a small sample of data. The use of a Markov chain Monte Carlo (MCMC) process for data-adaptive partitioning of an SRP is discussed in more detail in this chapter. Sections 6.2, 6.3 and 6.4 are based on Sainudiin et al. (2013). Sections 6.5, 6.7, and 6.6 discuss further research into some aspects of the MCMC sampler not covered in detail in Sainudiin et al. (2013): mixing, assessing convergence, and sampling from the chain. The result of this research is a 'semi-automatic' SRP MCMC process that includes a method to find suitable well-dispersed initial states for multiple Metropolis-Hastings Markov chains and to monitor the chains so that sampling can be automatically initiated once convergence has been diagnosed. Alternative MCMC samplers are discussed in Section 6.8.

The aim of data-adaptive partitioning of an SRP with root box $\boldsymbol{x}_\rho$ using an MCMC process is to obtain a sample mean estimate of the Bayesian posterior expectation of the density estimate based on a finite subspace of ${}^\circ\mathbb{S}_{0:\infty}$, the space of all SRPs with root box $\boldsymbol{x}_\rho$, i.e., the target is the posterior distribution of SRP histogram density estimates with root box x_ρ given realised values of the sample data ${}^nX$. The sample mean is an RMRP density estimate.

6.2 The posterior distribution

Given realisations of the sample data $x_1^\star, \ldots, x_n^\star$, let π be the posterior distribution that is proportional to the product of the likelihood of the data $x_1^\star, \ldots, x_n^\star$ given SRP ${}^\circ s$ and the prior probability of ${}^\circ s$.

The likelihood of the data given SRP ${}^\circ s$ is approximated by the maximum likelihood value from the histogram $\hat{f}_n$ with bins given by the partition $\boldsymbol{x}_{\mathbb{L}({}^\circ s)}$ of the root box of ${}^\circ s$:

$$\begin{aligned} \Pr\{x_1^\star, \ldots, x_n^\star | {}^\circ s\} &= \Pr\{x_1^\star | {}^\circ s\} \Pr\{x_2^\star | {}^\circ s\} \cdots \Pr\{x_{n-1}^\star | {}^\circ s\} \Pr\{x_n^\star | {}^\circ s\} \\ &\approx \mathcal{L}\left(\hat{f}_n\right) \ , \end{aligned}$$

where $\mathcal{L}\left(\hat{f}_n\right) = \prod_{\rho\nu \in \mathbb{L}(\,^\circ s)} \left(\frac{\#\boldsymbol{x}_{\rho\nu}}{n\mathrm{vol}\,(\boldsymbol{x}_{\rho\nu})}\right)^{\#\boldsymbol{x}_{\rho\nu}}$ (see Equation (3.4)). Then,

$$\begin{aligned} \pi(\,^\circ s) &:= \Pr\{\,^\circ s | x_1^\star, \ldots, x_n^\star\} \\ &\propto \Pr\{x_1^\star, \ldots, x_n^\star |\,^\circ s\} \Pr\{\,^\circ s\} \ , \end{aligned}$$

and

$$\Pr\{x_1^\star, \ldots, x_n^\star |\,^\circ s\} \Pr\{\,^\circ s\} \approx \mathcal{L}\left(\hat{f}_n\right) \Pr\{\,^\circ s\} \ .$$

The prior probabilities should decrease as the partition size increases in order to penalise large partitions. An $\{a_k\}$-*penalised uninformative proper Catalan prior* that assigns states in $^\circ\mathbb{S}_k$ with probability $\frac{a_k}{a}$ and distributes this mass uniformly over $^\circ\mathbb{S}_k$ is given by:

$$\Pr\{\,^\circ s\} = \sum_{k=0}^{\infty} \mathbb{1}_{\,^\circ\mathbb{S}_k}(\,^\circ s) \frac{a_k}{aC_k} \ , \tag{6.1}$$

where $\{a_k\}$ for $k = 1, 2, \ldots$ is any decreasing sequence of positive real numbers such that $\sum_{k=1}^{\infty} a_k = a < \infty$. The MCMC algorithm implemented in this thesis uses a prior obtained from the sequence of $a_k = \frac{1}{C_k}$. Given a, such a *natural Catalan prior* is given by:

$$\Pr\{\,^\circ s\} = \sum_{k=0}^{\infty} \mathbb{1}_{\,^\circ\mathbb{S}_k}(\,^\circ s) \frac{1}{aC_k^2} \ . \tag{6.2}$$

Thus, the posterior distribution on $^\circ\mathbb{S}_{0:\infty}$, up to proportionality, is given by

$$\pi(\,^\circ s) \propto \mathcal{L}\left(\hat{f}_n\right) \cdot \sum_{k=0}^{\infty} \mathbb{1}_{\,^\circ\mathbb{S}_k}(\,^\circ s) \frac{1}{aC_k^2} \ . \tag{6.3}$$

Sainudiin et al. (2013) use $a = 2 + 4\pi/3^{5/2}$. In the Metropolis-Hastings MCMC algorithm described below it is not necessary to use a normalised prior and no value for a is specified.

6.3 A finite state space

Section 4.3.2 describes limits on the leaf nodes $\mathbb{L}(\,^\circ s)$ of an SRP $^\circ s$ that may be considered to be splittable. These limits restrict the state space of SRPs that can be visited by a Markov chain. Limits imposed by the computer representation of floating point numbers are not optional in any computerised implementation of the MCMC process. These limits will result in a finite state space because only a certain number of recursive bisections within any box can occur before either the volume of the resulting box becomes too small to be computer-representable, or the interval equal to the first widest coordinate of the box to be bisected cannot be bisected

into two non-overlapping intervals (see Appendix C). A limit $\underline{\#}$ on the minimum number of data points associated with any non-empty leaf node in the SRP may or may not be applied. The effect of the computer-implementation limits or a particular value for $\underline{\#}$ on the allowable SRP states will depend on the sample data ${}^{n}X$ and cannot usually be determined before the partitioning takes place. These limits are referred to as 'data-dependent' limits.

The SRP state space may also be restricted by imposing 'hard' limits on the tree structure. Such limits include an upper limit $\overline{m}$ on the number of leaves so that $|\mathbb{L}({}^{\circ}s)| \leq \overline{m}$ for any SRP ${}^{\circ}s$ in the chain. This limits the total size of the partition exactly as described in the context of RPQs in Chapter 5. Alternatively, an upper limit $\bar{d}$ could be imposed on the depth in the tree of any leaf node in the SRP. The values for $\overline{m}$ and $\bar{d}$ may be set with reference to the total number of data points $n = \#\boldsymbol{x}_\rho$ associated with the root box $\boldsymbol{x}_\rho$ of the SRP and the effect of these limits on the allowable SRP states is known before any partitioning takes place. These limits are referred to as 'n-dependent' limits.

Let ${}^{\circ}\tilde{\mathbb{S}} \subset {}^{\circ}\mathbb{S}_{0:\infty}$ be a finite state space of SRPs defined by imposing data-dependent limits and, possibly, n-dependent limits, on ${}^{\circ}\mathbb{S}_{0:\infty}$. An example of such a restricted state is shown in Figure C.3 in Appendix C.

A restricted state space ${}^{\circ}\tilde{\mathbb{S}}$ for an MCMC algorithm must ensure that the chain is irreducible. This can be achieved by requiring that requires that:

- ${}^{\circ}\mathbb{S}_{0:0} \subseteq {}^{\circ}\tilde{\mathbb{S}}$; and
- any SRP state in ${}^{\circ}s \in {}^{\circ}\tilde{\mathbb{S}}$ can be reached from the unpartitioned (single-leaf) SRP ${}^{\circ}s \in {}^{\circ}\mathbb{S}_{0:0}$ by a sequence of selective splits and, conversely, the unpartitioned (single-leaf) SRP ${}^{\circ}s \in {}^{\circ}\mathbb{S}_{0:0}$ can be reached from any ${}^{\circ}s \in {}^{\circ}\tilde{\mathbb{S}}$ by a sequence of merges.

The computer-implementation limits described above give a restricted state space ${}^{\circ}\tilde{\mathbb{S}}$ that meets these conditions, as does a lower limit $\underline{\#}$ on the number of data points associated with a non-empty leaf node, provided that $0 < \underline{\#} \leq n = \#\boldsymbol{x}_\rho$. A limit on the maximum number of leaves $\overline{m} \geq 1$ and a limit on the maximum depth of a leaf node $\bar{d}_n \geq 1$ will also give ${}^{\circ}\tilde{\mathbb{S}}$ meeting these conditions for irreducibility.

6.4 A Metropolis-Hastings MCMC sampler

The SRP MCMC algorithm used for this thesis, except if explicitly stated otherwise, is the Metropolis-Hastings algorithm described in this section.

6.4.1 Base Markov chain

This section describes a *stay-split-merge* base Markov chain $\{Y(t)\}_{t\in\mathbb{Z}_+}$ on the finite state space ${}^{\circ}\tilde{\mathbb{S}}$. Let ${}^{\circ}s$ be the current state of the SRP. A proposal to stay in the current state has positive probability ς. A proposal to move to another state has probability $1-\varsigma$. If a move

is chosen, it can be a bisection of one of the splittable leaf nodes $\rho \mathsf{v} \in \mathbb{L}^{\triangledown}(\,^{\circ}s)$ or a reunion of one of the cherry nodes $\rho \mathsf{v} \in \mathbb{C}(\,^{\circ}s)$ with equal probability $\frac{1-\varsigma}{2}$. If a bisection is chosen, each splittable leaf node in the current state $^{\circ}s$ has an equal probability $\frac{1-\varsigma}{2|\mathbb{L}^{\triangledown}(\,^{\circ}s)|}$ of being bisected. Similarly, if a reunion is chosen, each cherry node in $^{\circ}s$ has an equal probability $\frac{1-\varsigma}{2|\mathbb{C}(\,^{\circ}s)|}$ of having its sibling nodes reunited to itself. The transition probabilities between any two states $^{\circ}s,\, ^{\circ}s' \in {}^{\circ}\tilde{\mathbb{S}}$ are:

$$Q(\,^{\circ}s,\, ^{\circ}s') = \begin{cases} \dfrac{1-\varsigma}{2|\mathbb{L}^{\triangledown}(\,^{\circ}s)|} & \text{if a node } \rho \mathsf{v} \in \mathbb{L}^{\triangledown}(\,^{\circ}s) \text{ can be split once to get } {}^{\circ}s' \\ \dfrac{1-\varsigma}{2|\mathbb{C}(\,^{\circ}s)|} & \text{if a node } \rho \mathsf{v} \in \mathbb{C}(\,^{\circ}s) \;\; {}^{\circ}s \text{ can be reunited once to get } {}^{\circ}s' \\ \varsigma & \text{if } {}^{\circ}s = {}^{\circ}s' \\ 0 & \text{otherwise .} \end{cases} \tag{6.4}$$

Any SRP state $^{\circ}s \in {}^{\circ}\tilde{\mathbb{S}}$ can be reached from the unpartitioned (single-leaf) SRP $^{\circ}s \in {}^{\circ}\mathbb{S}_{0:0}$ by a sequence of selective splits, and the unpartitioned (single-leaf) SRP $^{\circ}s \in {}^{\circ}\mathbb{S}_{0:0}$ can be reached by a sequence of selective merges from any state $^{\circ}s \in {}^{\circ}\tilde{\mathbb{S}}$. Therefore the chain $\{Y(t)\}_{t\in\mathbb{Z}_+}$ on a finite state space ${}^{\circ}\tilde{\mathbb{S}}$ is irreducible. The chain is also aperiodic since there is a positive probability ς of staying in the current state. Therefore, the base chain has a unique stationary distribution (Levin et al. 2009, chap. 4).

The MCMC SRP results discussed in this thesis are obtained using the stay-split-merge base Markov chain with $\varsigma = 10^{-6}$.

6.4.2 The Metropolis-Hastings Markov chain

Using the irreducible and aperiodic base chain $\{Y(t)\}_{t\in\mathbb{Z}_+}$ on the finite state space ${}^{\circ}\tilde{\mathbb{S}}$ with transition matrix Q in Equation 6.4 and the posterior distribution π given in Equation 6.3, the following transition probabilities give a Metropolis-Hastings chain $\{S(t)\}_{t\in\mathbb{Z}_+}$ on ${}^{\circ}\tilde{\mathbb{S}}$ with π truncated to ${}^{\circ}\tilde{\mathbb{S}}$ as its stationary distribution:

$$P(\,^{\circ}s,\, ^{\circ}s') = \begin{cases} Q(\,^{\circ}s,\, ^{\circ}s')a(\,^{\circ}s,\, ^{\circ}s') & \text{if } {}^{\circ}s \text{ leads to } {}^{\circ}s' \text{ by a single split or merge} \\ 1 - \displaystyle\sum_{z \in {}^{\circ}\tilde{\mathbb{S}} \setminus {}^{\circ}s} Q(\,^{\circ}s, z)a(\,^{\circ}s, z) & \text{if } {}^{\circ}s' = {}^{\circ}s \\ 0 & \text{otherwise ,} \end{cases}$$

where the acceptance probability is

$$a(\,^{\circ}s,\,^{\circ}s') := \min\left\{1, \frac{\pi(\,^{\circ}s')Q(\,^{\circ}s',\,^{\circ}s)}{\pi(\,^{\circ}s)Q(\,^{\circ}s,\,^{\circ}s')}\right\} \quad .$$

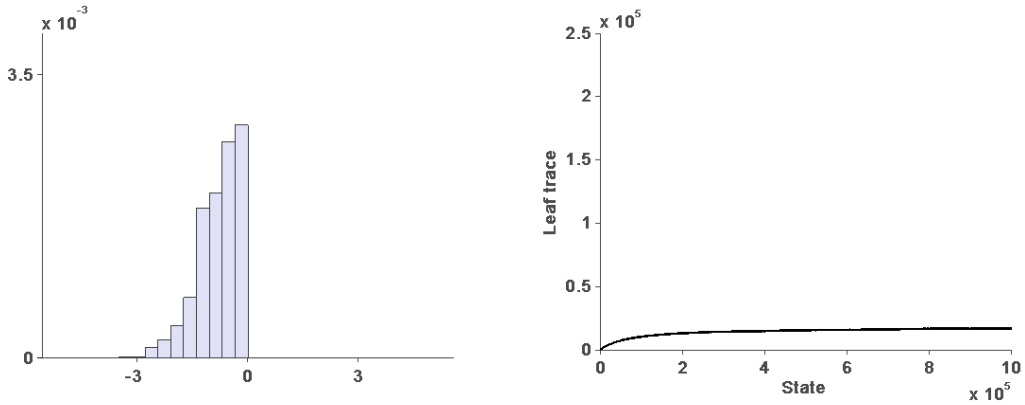
6.5 Exploring the state space with the Metropolis-Hastings Markov chain

6.5.1 Mixing

Details of the behaviour of the Metropolis-Hastings Markov chain described here are discussed in Appendix E. The stay-split-merge base chain means that mixing will be slow: each state is only one split or merge away from the previous one. As the simple example in Section 4.5 illustrated, the chain is also susceptible to getting at least temporarily trapped in a small sub-space of the total state space.

An example of very poor mixing is illustrated in Figure 6.1. The sample data is $n = 1 \times 10^7$ data points simulated from a 6-d multivariate Gaussian distribution and the SRP has root box $[-5.573, 5.573]^6$. Figure 6.1(a) shows an unnormalised slice (see Algorithm 3.7, Section 3.5.2) parallel to the first coordinate through the state immediately after transition $t = 1,000,000$ in the chain. A trace of the number of leaves in the SRP against the transition index t is shown in Figure 6.1(b). The vertical scale on this figure is chosen to contrast the very low number of leaves in the SRP throughout the 1,000,000 states in the chain to the number of leaves in the SRP in the better-mixing chain discussed in Section 6.5.2 below. Detailed state-by-state logs showed that in all these states most of the partitioning took place in the left half of the root box. No more than very shallow partitioning in the right side of the root box was achieved in any state in the chain up to $t = 1,000,000$ so that most of the states in the chain up to that point had a similar non-symmetric overall shape to that illustrated in the slice shown in Figure 6.1(a).

The MCMC algorithm described here is likely to perform relatively badly with data with a strongly symmetrical structure. A chain is especially likely to get trapped in very shallowly partitioned states when the sample data is multivariate and appears to be evenly distributed within each box in the partition until not just one but several successive bisections have been made. In the case of the chain shown in Figure 6.1, it is only after one split on each coordinate of the root box (i.e., six splits not reversed by merges) that the Gaussian data does not appear to be almost symmetrically distributed when each immediate next transition is proposed. However, if the partitioning has become uneven in the sense that one major branch of the SRP tree is trapped in a relatively unsplit shallow state but another has been more thoroughly (deeply) split, then this unevenness can persist for a very considerable number of transitions in the chain (see Section E.5 of Appendix E).



(a) Slice of MCMC state at $t = 1,000,000$ (16,696 leaves).

(b) Leaf trace to $t = 1,000,000$.

Figure 6.1: MCMC partitioning with data drawn from a 6-d Gaussian distribution.

Similar mixing problems can occur in less exaggerated form with less symmetrically distributed sample data, and the small-step transitions given by stay-split-merge base chain will inevitably mean slow mixing even in the best circumstances. The advantages of the Metropolis-Hastings algorithm using the stay-split-merge base chain are that it is relatively easy to implement and, exactly because the changes from one state to another are only small, the acceptance ratio is reasonably high.

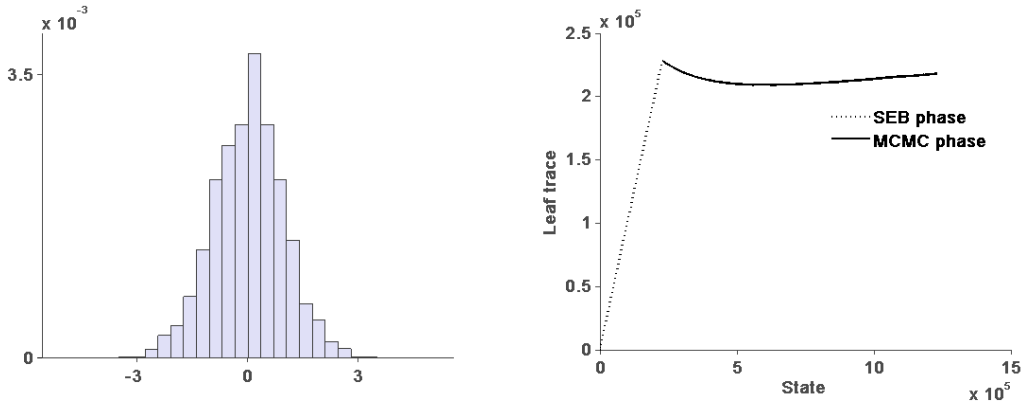
6.5.2 Initial state

If the Metropolis-Hastings chain $\{S(t)\}$ is initialised too far from the states with high posterior mass then the mixing time can be prohibitively large. An example of this is illustrated in Figure 6.1. A solution is to use an SEB-based RPQ (possibly combined with a 'carving queue', as described in Section 5.4), to obtain a better initial state for the chain. The relative strengths and weaknesses of the two methods tend to complement each other so that, in combination, some of the worst aspects of both may be mitigated:

- It is hard to find appropriate parameters to control an SEB-based RPQ and prevent undersmoothing but the MCMC has the opportunity to reunite any over-split (over-refined) areas of the partition that may be present in the initial state;

- The SEB-based RPQ does naturally what the MCMC is reluctant to do — repeated splitting of large areas of apparently uniform density. When the SEB-based RPQ is used on its own, this is a disadvantage (see Section 4.5) but when the SEB-based RPQ is used to obtain an initial state for an MCMC it can 'crack open' the sample very efficiently and give the MCMC a chance to 'see' more of the underlying structure in the data.

Figure 6.2 shows the effect of using a state from an initial SEB-based RPQ phase as the starting state for the MCMC process for the same sample data as was used for Figure 6.1. The 1-d slice (again an unnormalised slice parallel to the first data coordinate taken from the state at $t = 1,000,000$ in the chain) shown in Figure 6.2(a) is a considerable improvement on that shown in Figure 6.1(a). Figure 6.2(b) shows the leaf trace from both the SEB RPQ phase and the MCMC phase combined (up to $t = 1,000,000$ in the chain). Again, the contrast to Figure 6.1(a) demonstrates clearly the very limited partitioning achieved when the chain was started from the root node.



(a) Slice of MCMC state.

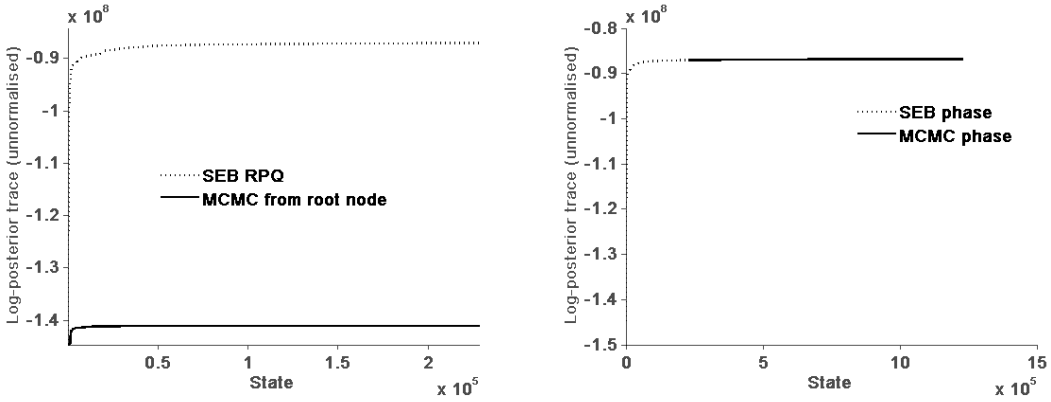
(b) Combined leaf trace to $t = 1,000,000$.

Figure 6.2: MCMC partitioning after initial SEB phase, data drawn from a 6-d Gaussian distribution.

Figure 6.3 shows trace plots of the unnormalised log-posterior mass of the histogram density estimate derived from the SRP at each state in the partitioning process. The posterior mass of an RMRP density estimate $\hat{f}_n$ based on an SRP $\circ s$ with $\boldsymbol{x}_\rho = n$, $m = |\mathbb{L}(\circ s)|$ leaves, and $k = m - 1$ splits is $\pi(\circ s) \propto \frac{\mathcal{L}(\hat{f}_n)}{C_k^2}$ (Equation (6.3)). The unnormalised posterior values can be used for the trace plots because only the relative posterior mass of the same SRP in different states is of interest.

Figure 6.3(a) compares the log-posterior trace for the initial states in the chain started from the root node (Figure 6.1) with the log-posterior trace for states in a short SEB RPQ. This shows clearly how ineffective the stay-split-merge Metropolis-Hastings algorithm can be with challenging data, such as the simulations from the 6-d Gaussian distribution, if it is started from a state with very low posterior mass. Figure 6.3(b) shows the total log-posterior trace for an SRP associated with the same sample data and initially partitioned using a short SEB RPQ to get a starting state for a subsequent 1,000,000-state chain.

Further research on this topic could consider alternative methods for finding a suitable starting state for an SRP Metropolis-Hastings Markov chain.



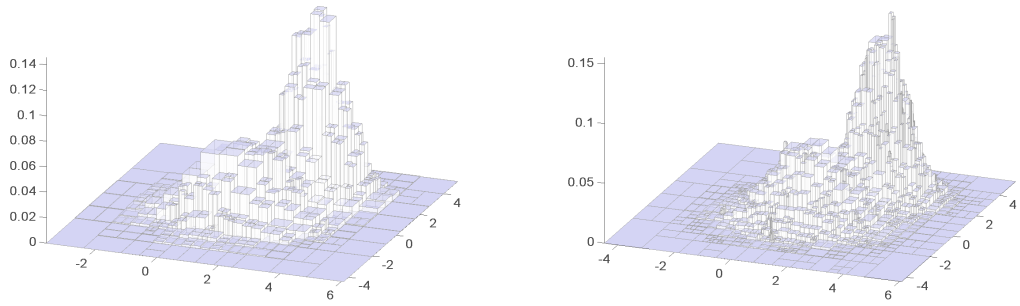
(a) Initial SEB phase compared with MCMC from root node.

(b) Combined log-posterior trace to $t = 1,000,000$.

Figure 6.3: Un-normalised log-posterior mass traces, data drawn from a 6-d Gaussian distribution.

6.5.3 The effect of the sample size

The influence of the prior on the behaviour of the Metropolis-Hastings Markov chain depends on the magnitude of the total sample size $n = \#\boldsymbol{x}_\rho$, the number of data points associated with the root node ρ of the SRP (see Appendix E). As n increases the natural Catalan prior (Section 6.2) becomes relatively weaker as the effect of the likelihood on the acceptance probability becomes stronger. In general the chain will explore more deeply-split states when n is larger.



(a) Sample size $n = 10,000$.

(b) Sample size $n = 100,000$.

Figure 6.4: The effect of increasing sample size on the complexity of the averaged RMRP estimate.

Figure 6.4 illustrates this, showing the RMRP density estimate created by averaging 100 MCMC SRP histogram samples for $n = 10,000$, and $n = 100,000$ data points all drawn from example Density I, $d = 2$ (see Appendix B). The number of leaves in the average increases

from 618 for $n = 10,000$ to 1,628 for $n = 100,000$.

The depth of exploration could be controlled by a extending the natural Catalan prior described in Section 6.2 to a family of priors indexed by a temperature parameter. A 'hotter' prior would be weaker and allow the chain to explore deeper (more split) states for the same sample size n. However, it is difficult to select the optimal temperature when the true distribution of the data is unknown without resorting to computationally intensive smoothing techniques such as cross-validation. This is an important topic for future research to improve MCMC methods for SRPs.

6.6 Convergence

An irreducible aperiodic MCMC sequence converges asymptotically to its stationary distribution. Ideally, samples from the chain would be taken once that stationary distribution is reached. In practice, assessing convergence means determining whether the chain has been run for long enough for states in the chain to be "approximate draws from the posterior distribution of interest" (Sorensen and Gianola 2002, p. 541). Assessing convergence is a problematic issue for any form of MCMC sampling. The methods used range from informal monitoring of the progress of the chain, often using trace plots, to an extensive battery of more formal convergence diagnostic calculations and graphical techniques (Cowles and Carlin 1996). This section describes the methods used in this thesis to assess convergence of chains of SRP states to the target, the posterior distribution of SRP histogram density estimates.

6.6.1 Summarising histogram state

Most of the conventional methods for assessing convergence, whether they involve informal monitoring of the chain with trace plots or more formal diagnostic calculations, use univariate or low-dimensional multivariate scalar values to summarise each state in the chain (see, for example, Cowles and Carlin (1996), Brooks (1998), and Brooks and Gelman (1998)). Ideally, realised states in the chain themselves can be used. The posterior distribution of interest in this chapter is a posterior distribution of SRP histograms represented as RMRPs. The RMRPs are very high-dimensional and the actual states in the chain cannot easily be used, unsummarised, in assessing convergence. Let $V : {}^{\circ}\tilde{\mathbb{S}} \to \mathbb{R}^d$ be some mapping of SRP states in ${}^{\circ}\tilde{\mathbb{S}}$ to values in $\mathbb{R}^d$ where d is sufficiently low for $V({}^{\circ}s)$ to be useful for assessing convergence. Intuitively, an ideal V would have two additional properties:

- $V({}^{\circ}s)$ would be computationally cheap to calculate for all ${}^{\circ}s \in {}^{\circ}\tilde{\mathbb{S}}$, so that assessing convergence is practically feasible; and
- $V({}^{\circ}s^{(1)}) - V({}^{\circ}s^{(2)}) \propto |{}^{\circ}s^{(1)} - {}^{\circ}s^{(2)}|$ where $|{}^{\circ}s^{(1)} - {}^{\circ}s^{(2)}|$ is some measure of the distance between SRP states ${}^{\circ}s^{(1)}$ and ${}^{\circ}s^{(2)}$, so that a decrease (increase) in $V({}^{\circ}s^{(1)}) - V({}^{\circ}s^{(2)})$ indicates a decrease (increase) in the distance between ${}^{\circ}s^{(1)}$ and ${}^{\circ}s^{(2)}$.

In practice it is difficult to find scalar summaries that have both of these properties. The following univariate scalar summary values of an SRP ${}^{\circ}s$ are considered in this chapter:

- The number of leaves $|\mathbb{L}({}^{\circ}s)|$.
- The number of cherries $|\mathbb{C}({}^{\circ}s)|$.
- The average leaf node depth $\frac{\sum_{\rho v \in \mathbb{L}({}^{\circ}s)} \mathsf{d}_{\rho v}}{|\mathbb{L}({}^{\circ}s)|}$.

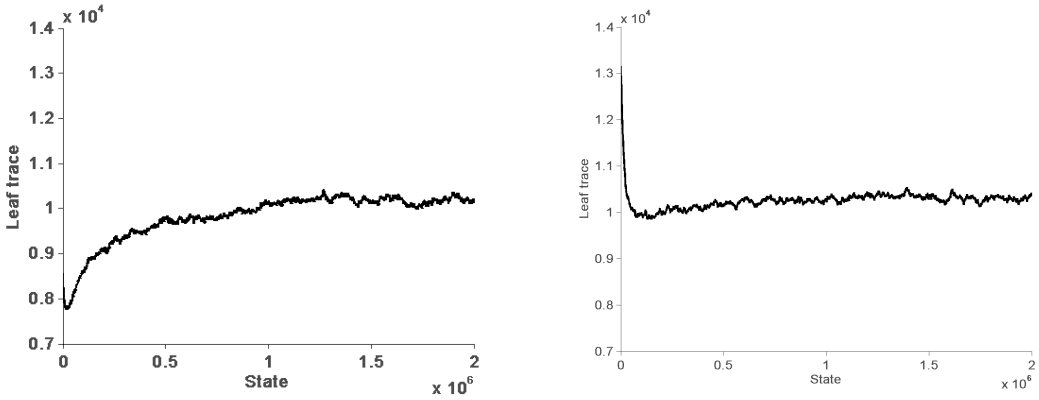
All of these scalars are computationally cheap to calculate for the Metropolis-Hastings Markov chain with a stay-split-merge base chain as described above because the scalar values can easily be updated for each change in state in a chain. None of them provides a good measure of the distance between SRP states because, for each one, many different SRP states can yield the same scalar value (the mapping $V : {}^{\circ}\tilde{\mathbb{S}} \to \mathbb{R}$ is non-injective): two very different states can appear to be similar when measured by the difference in their numbers of leaves, numbers of cherries, or average leaf depths. However, thus far in the research for this thesis, no scalars have been found that provide a better measure of the difference between states but are also feasible (cheap enough to calculate) in the context of the very long chains necessitated by the slow-mixing MCMC algorithm and very large state space of SRPs.

Given leaves $m = |\mathbb{L}({}^{\circ}s)|$, the maximum possible number of cherries (in a close-to-perfect tree, where the depths of the leaf nodes differ by a maximum of one) is $\lfloor \frac{m}{2} \rfloor$ and the minimum number of cherries (in the most unbalanced possible tree with minimum leaf depth 1 and maximum leaf depth $m-1$) is 1. The number of leaves and the number of cherries together therefore provide additional information about the shape of an RP tree. The higher the ratio of leaves to cherries, the more unbalanced the tree (but again many different unbalanced shapes will yield the same leaves:cherries ratio). The number of leaves together with the average leaf node depth provides similar additional information about the tree shape. One of the questions addressed in this section is which combination of scalar values seems to provide the most useful information with which to assess convergence.

6.6.2 Assessing convergence using trace plots

Trace plots from individual chains do not usually give enough information to be reliable tools for assessing convergence. This is particularly true when low dimensional summaries of the higher-dimensional random variable being simulated are traced (Gelman and Rubin 1992). As is discussed above, the low-dimensional scalar values suitable for use in such trace plots when the random variable is an SRP only provide a very highly-compressed summary of the SRP state. In addition, when the true density of the sample data ${}^{n}X$ is entirely unknown, it is impossible to assess the trace plot by comparing the plot with some 'reasonable' or expected range of values.

Figure 6.5 shows leaf traces from two MCMC processes. In each case it is hard to tell whether the point at which the leaf traces stabilises indicates that convergence has been achieved or simply that the chain has become trapped.



(a) Chain 1.

(b) Chain 2.

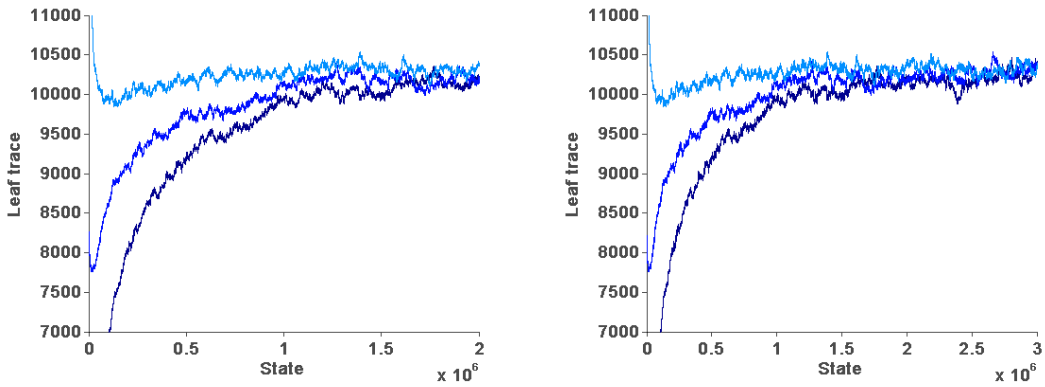
Figure 6.5: Leaf traces for two chains to $t = 2,000,000$.

Although summaries such as the number of leaves, number of cherries, or average leaf depth discussed above in Section 6.6.1 do not provide enough information to reliably affirm convergence, comparing trace plots of these values from multiple chains can sometimes indicate that convergence has *not* taken place. Similar values for the leaves, cherries or average depth scalar summaries do not necessarily indicate similar SRP states, but very dissimilar SRP states will give dissimilar values of the leaves and cherries scalar summaries (but possibly not average leaf depth). Mixing problems and chains that are temporarily trapped can often be identified by comparing trace plots for multiple chains.

The two chains in Figure 6.5 are in fact chains for the same SRP started from different initial states. Figure 6.6 shows leaf traces for both these chains together with the trace for a third chain, also for same SRP, started from a third initial state. Figure 6.6(a) shows the traces over the same length of chain as in Figure 6.5 (the first 2,000,000 transitions). There are persistent differences between the number of leaves in the three SRPs until about 1,500,000 transitions have taken place. Figure 6.6(b) extends the traces over longer chains (to $t = 3,000,000$), showing that after $t = 1,500,000$ all three chains begin to move mainly within a subset of states with a similar range of numbers of leaves.

6.6.3 Obtaining more than one initial state

The advantages of using a short SEB-based RPQ to obtain the initial state for a subsequent MCMC process were discussed in Section 6.5.2. Figure 6.3(b) showed a plot of the unnormalised log-posterior mass at each state in both an initial SEB RPQ and then a Markov chain for an SRP with data simulated from a 6-d Gaussian distribution. Most of the total changes in log-



(a) Three chains to $t = 2,000,000$.

(b) Three chains to $t = 3,000,000$.

Figure 6.6: Leaf traces for multiple chains.

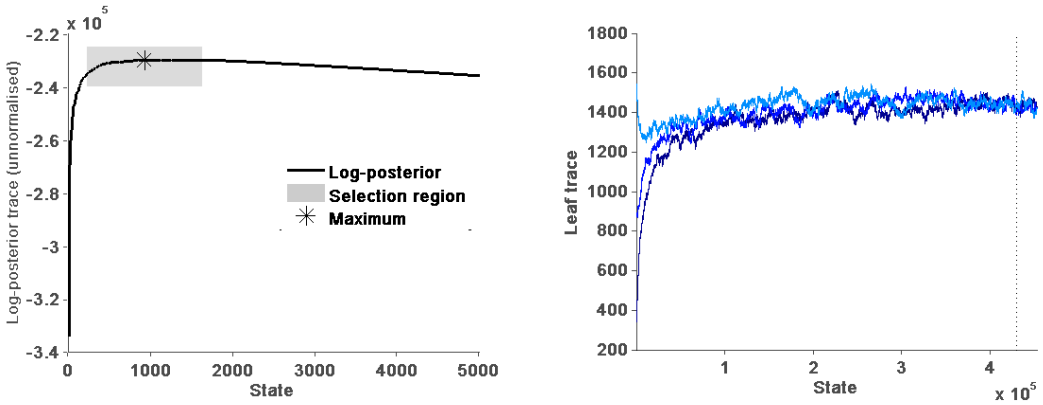
posterior mass (from initial state to the maximum over all states in the process) take place in the first states in the SEB RPQ. This is typical of the effect of this two-phase process: the same pattern was found with every other set of structured sample data tested during the course of this thesis.

This suggests that more than one initial state can be found using the same initial SEB RPQ by selecting states from the RPQ sequence $\{S(k)\}_{k\in\mathbb{Z}_+}$ (see Section 5.2) once most of the total change in log-posterior mass has been achieved.

The heuristic method to obtain multiple MCMC initial states implemented for this thesis is described in Appendix F. The aim is to select well-spaced states from a sub-sequence of the total sequence of states $\{S(k)\}_{k\in\mathbb{Z}_+}$ obtained from an SEB RPQ (or a combined carving and SEB RPQ as described in Sections 5.4 and 6.5.2). The sub-sequence is chosen to include the state with the maximum log-posterior in the whole RPQ sequence and to start at the state at which some large proportion α (say, 95%) of that maximum log-posterior mass is first attained.

Figure 6.7(a) illustrates this, showing the trace of the (unnormalised) log-posterior over the whole SEB RPQ sequence, the point where the maximum log-posterior mass is attained, and the sub-sequence of states from which multiple initial states are chosen ("the selection region"). If the number of chains required is $c = 3$ then the three chosen initial states would be the maximum log-posterior point $*$ shown in Figure 6.7(a) (933 leaves), the state with the fewest number of leaves (234) in the selection region, and the state with the largest number of leaves (1,632) in the selection region. Figure 6.7(b) shows the trace plots of the numbers of leaves in three chains started from these states. Eventually all three chains move between states with around 1,400–1,450 leaves. These traces relate to an SRP MCMC process to make a density estimate using a sample of $n = 50,000$ data points simulated from example Density I, $d = 3$ (see Appendix B).

The heuristic method described here can be used with a wide range of different values of



(a) Selection region.

(b) Leaf trace for three chains.

Figure 6.7: Selecting multiple initial states.

c. In most of the examples used in this thesis three chains are used, but larger numbers of initial states can easily be found.

Other heuristic methods could be used for finding multiple initial states. Any method that selects initial states with reasonably high posterior mass that are well-diversified (different to each other) would probably be satisfactory. If the initial states are too far from the states with high posterior mass the mixing time will be prohibitively large (see Section 6.5.2), and if the initial states are not well-diversified then convergence can be falsely diagnosed too early simply because the chains are behaving similarly. This is especially important if some form of automated method is used to diagnose convergence.

The time taken to find initial states using this process is a very small proportion of the total time required for the entire SRP MCMC process and is much lower than the amount of time saved in waiting for convergence to be achieved.

6.6.4 The Gelman-Rubin potential scale reduction factor as a convergence diagnostic

Sainudiin et al. (2013) describe a convergence heuristic for SRP chains based on the potential scale reduction factor (PSRF) $\widehat{R}_{\text{variance}}$ of Gelman and Rubin (1992). This is a measure of the ratio of the estimate of the current variance of v across all the chains to the estimated within-chain variance of v where v is the scalar summary used to assess convergence. The within-sequence variance is less than the variance across all the chains but, as $t \to \infty$ and the chains explore more of the target distribution, $\widehat{R}_{\text{variance}} \to 1$. A value of $\widehat{R}$ close to 1 should indicate that each of the chains is close to the target distribution, subject to ensuring that the variance across the chains and the within-chain variance measure have also both stabilised (Brooks and Gelman 1998).

The theory of the PSRF $\widehat{R}_{\text{variance}}$ assumes that v is approximately normally distributed,

either because the target is approximately Normal or because v is calculated as some appropriately transformed statistic of the random variables being simulated (Gelman and Rubin 1992). In the case of chains of SRP states it may not be reasonable to assume that scalar summaries such as number of leaves, number of cherries, and average leaf depth, are appropriate for use with the $\widehat{R}_{\text{variance}}$ convergence heuristic (although with a sufficiently large sample size the posterior should become sufficiently concentrated for this assumption to be valid).

6.6.5 The interval-based $\widehat{R}$ convergence diagnostic

Brooks and Gelman (1998) describe a more general $\widehat{R}$ diagnostic that uses an interval rather than the sample variance to measure the variability of the scalar v. A $100(1-\alpha)\%$ interval can be calculated for each chain by calculating the $100\frac{\alpha}{2}$ percentile and $100\left(1-\frac{\alpha}{2}\right)$ percentile points for a set of scalar values summarising a set of states in the chain. This gives c interval-width estimates of within-chain variability if c chains are used to assess convergence. Similarly, an interval-width estimate of variability across the c chains can be calculated by combining the sets of v from all chains. This is the *total-chain interval*. The diagnostic measure $\widehat{R}$ is then defined as

$$\widehat{R}_{\text{interval}} = \frac{\text{width of total-chain interval}}{\text{mean width of the within-chain intervals}} \ .$$

$\widehat{R}_{\text{interval}}$ is the diagnostic measure used to assess convergence in this thesis. Henceforth the subscript 'interval' is dropped and the original variance-based $\widehat{R}$ of Gelman and Rubin (1992) is referred to, where necessary, as $\widehat{R}_{\text{variance}}$.

Gelman and Rubin (1992) treated the convergence diagnostic as a one-off calculation used to check an assumption of convergence. Some number of states n_t is chosen, at the end of which it is assumed that convergence will have taken place. Each chain is run to generate a total sequence of $2n_t$ states and only the second half of the sequence, the last n_t states, is used to calculate $\widehat{R}_{\text{variance}}$. Brooks and Gelman (1998) discuss this and an alternative 'iterated graphical approach' that divides each sequence into batches and calculates $\widehat{R}$ (or $\widehat{R}_{\text{variance}}$) using the second half of the sub-sequence for each batch and then plots the sequence of batch $\widehat{R}$ to monitor the progression toward convergence as the sequences get longer.

The approach used to assess convergence in this thesis is an adaptation of the iterated graphical approach described in Brooks and Gelman (1998). The interval-based $\widehat{R}$ is calculated over a moving sub-sequence of states that may become longer as the total sequence length increases. The total length of chain depends on when convergence is diagnosed and how the SRP states are sampled once this has been achieved. The method used to calculate the convergence diagnostic $\widehat{R}$ for the MCMC process developed in this thesis is described in detail in Appendix G.

6.7 Automated sampling using a convergence diagnostic

The time taken to run a multiple-chain SRP MCMC process until convergence seems to have taken place using the implementation developed for this thesis is of the order of hours or days for reasonably large samples of multivariate data, and increases with both the sample size and the number of dimensions. Convergence diagnostics such as the interval-based $\widehat{R}$ of Brooks and Gelman (1998) are therefore of particular interest: it is at best inconvenient, more usually impractical, to perform several trial-and-error runs of an entire MCMC process in order to establish a suitable number of states to use to burn-in the chain(s) before sampling. It is more practical to try to use some automated method to determine when convergence has taken place and then obtain the required samples.

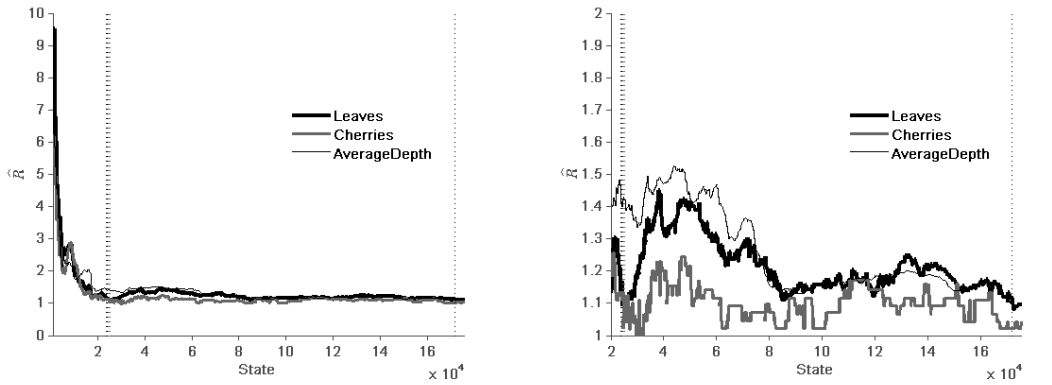
However, Brooks and Gelman (1998) point out that convergence should not just be assessed by looking for values of $\widehat{R}$ close enough to 1. Traces of the actual variability measures (for example, the mean within-chain interval width and total-chain interval width) must be considered as well. Convergence cannot be said to have taken place until the sequences of these values have also stabilised. In addition, trace plots of the sequences of scalar values for each chain should also be reviewed (for example, to check that sufficiently diversified initial values have been used for each chain). It is relatively easy to implement an automated sampler where sampling is triggered when the values of the interval-based $\widehat{R}$ diagnostic measure for various different scalar values are close enough to 1. It is more complicated to also automate ways to take into account all these other, sometimes more subjective, factors.

The approach adopted in this thesis is to use an automated sampling method where sampling is triggered by the values of the interval-based $\widehat{R}$ diagnostic measure for various different scalar values. Enough additional information is also produced by the process for the other useful trace plots described above to be produced. These trace plots should be examined to assess whether the final density estimate produced by the process should be accepted.

6.7.1 Using $\widehat{R}$ to diagnose convergence

The automated sampling method developed for this thesis looks for the transition T after which the value of $\widehat{R}$ is first near enough to 1 to indicate that each of the chains in the MCMC process may be close to the target distribution. The tolerance used for most of the examples and results given in this thesis is 0.1, i.e., a value of $\widehat{R} \leq 1.1$ is taken to indicate convergence. Examination of the various trace plots of the scalar summaries themselves, the within-chain interval widths and the total-chain interval widths, and selected SRP states in the chains carried out in testing for this thesis suggests that the tolerance of 0.1 is possibly too large if only a single scalar summary value (one sequence of $\widehat{R}$ values) is used, but is adequate when all three scalar summaries discussed here (the number of leaves, the number of cherries, and the average leaf depth) are considered.

The use of more than one scalar summary of the state of the random variable to assess MCMC convergence is in fact generally recommended (Cowles and Carlin 1996; Brooks and Gelman 1998)). Figure 6.8(a) shows the sequences of $\widehat{R}$ values for t =1,000–175,500 calculated using all three scalar summaries for an SRP MCMC process to make a density estimate using a sample of $n = 50,000$ data points simulated from example Density I, $d = 2$ (see Appendix B). The vertical dotted lines indicate where the value of $\widehat{R}(t)$ in each sequence $\{\widehat{R}(t)\}$ first falls below 1.10. Both $\{\widehat{R}_{\text{leaves}}(t)\}$ and $\{\widehat{R}_{\text{cherries}}(t)\}$ (the sequences using the number of leaves and number of cherries, respectively, as scalar values) temporarily fall below 1.10 relatively early but then rise again somewhat, while the $\{\widehat{R}_{\text{average depth}}(t)\}$ sequence takes considerably longer to drop below 1.10 for the first time. Figure 6.8(b) shows a close-up on the traces for t =20,000–175,500.



(a) Traces for t =1,000–175,500.

(b) Close-up on traces for t =20,000–175,500.

Figure 6.8: $\widehat{R}$ with different scalars summaries.

Appendix G discusses the use of multiple scalar summaries in more detail and concludes that none of the scalar summaries for SRP state discussed here is generally most or least sensitive to the convergence of the chains, nor is any of these scalars redundant in the sense of giving no further useful information in addition to that available from the other two. The automated sampling method developed for this thesis ends the burn-in period and starts sampling from the chains once the values of all three $\{\widehat{R}(t)\}$ sequences ($\{\widehat{R}_{\text{leaves}}(t)\}$, $\{\widehat{R}_{\text{cherries}}(t)\}$ and $\{\widehat{R}_{\text{average depth}}(t)\}$) are all below 1.10. No other tolerance values were tested in the course of this thesis but the effect of varying this tolerance should be included in further research into this automated sampling method.

6.7.2 Sampling

The aim of the sampling process is to obtain an estimate of the expectation of the posterior distribution of SRP histograms. The auto-correlation between the states in the sequence should be taken into account when the sampling scheme is determined. The sampling scheme specifies

which chains the samples are obtained from, how many SRP states in total to sample, and what thin-out value to use. The thin-out value is the number of transitions between the sampled states. A thin-out value of r ($r \geq 1$) means that every r^{th} state in a post-burn-in sequence $\{S(t)\}_{t>T}$ is sampled. If $r = 1$ the sequence is not thinned.

The states in the sequence $\{S(t)\}_{t\in\mathbb{Z}_+}$ given by a Metropolis-Hastings chain with the stay-split-merge base chain are very highly correlated: adjacent SRP states in the sequence differ by at most one split or merge. If the chain reaches its stationary distribution at transition T then the sequence $\{S(t)\}_{t>T}$ gives a *dependent* (auto-correlated) sample. The average of the states in this post-burn-in sequence converges almost surely to the expectation of the stationary distribution (Lee 2008) but a collection of sampled states taken over too short a length of chain will be very similar to each other. The sampling scheme should aim to ensure that the total length of chain (number of transitions) over which samples are taken is long enough to mitigate this problem.

If an *independent* sample of N realisations of a random variable simulated using an MCMC process is required, for example to estimate the posterior variance, then ideally this would be found by running N chains until each chain is burnt in and then sampling one state from each chain. Alternatively more than one state can be sampled from each of less than N chains, with enough transitions between the samples (a high enough thin-out value) to be able to consider them to be independent. Strictly, to get independent samples, the thin-out value should be the same as the burn-in time (Bolstad 2010, Chap. 7). Tierney (1994) suggests that the sampling scheme may aim to reduce, rather than eliminate, the dependence between the samples and use a lower thin-out value.

It is common practice to thin the samples even if the sample is only to be used to estimate the posterior mean (Link and Eaton 2012). Link and Eaton (2012) argue that unless it is actually necessary, thinning can be wasteful and gives a less precise estimate of the posterior mean than using all the samples. Thinning may still however be necessary or desirable if it is computationally difficult to base the average on the (larger) un-thinned sample.

In the implementation used in this thesis samples are taken from all c chains used to diagnose convergence, as Gelman and Rubin suggested when proposing the $\widehat{R}$ convergence diagnostic (Gelman and Rubin 1992). An alternative approach would be to only sample from one of the chains (letting the others stop once the burn-in period is over). The main argument in favour of sampling from a single very long chain is that this method "has a better chance of producing samples which properly represent the complete support of the target distribution" (Sorensen and Gianola 2002, p. 540). In the case of the Metropolis-Hastings MCMC algorithm used here, the stay-split-merge base chain means that it will take a single chain a very large number of transitions to move through the complete support of the posterior. Unfortunately many of the standard methods for estimating the post-burn-in run length required to adequately cover the posterior, using estimates of the posterior variance or

correlation between states (see, for example, Tierney (1994) and Sorensen and Gianola (2002, Chap. 12)), are of limited help when the random variable is very high-dimensional. There might be potential for using the same scalar summaries as are used to assess convergence in order to make some sort of estimate of a suitable number of transitions over which to spread the samples taken from each chain, although the variability of these summaries may severely underestimate the variability of the actual SRP state. There was not time during this thesis to develop this aspect of the process.

Two aspects of sampling were investigated for this thesis:

- The effect of the thin-out value and the number of SRP samples taken on the number of leaves in the final average RMRP, the time taken to obtain the final average, and the L_1 error of the final average against the true density.
- The effect of number of data points in the data sample ${}^{n}X$ associated with the SRP on the leaves in the final average RMRP, the time taken to obtain the final average, and the L_1 error of the final average against the true density.

The full results of these investigations are given in Appendix H and are summarised here.

The total number of transitions over which sampling takes place is referred to as the number of *sampling transitions*. Sampling N SRP states from one chain with thin-out r requires rN sampling transitions. The results in Appendix H suggest two relationships between the number of leaves in the average RMRP, the thin-out value, the number of SRP states sampled, and the computational cost of the sampling process as measured by the time taken:

- The number of samples taken has much more effect on the computational cost of sampling than does the total number of sampling transitions: for a fixed total number of sampling transitions, using a higher thin-out value can give dramatic increases in speed.
- The total number of sampling transitions can have a much stronger effect on the number of leaves in the final average than the number of samples taken: for a fixed total number of sampling transitions, using a higher thin-out value (sampling fewer states) may only give a small decrease in the number of leaves in the average. This is dimension dependent. The higher the dimensions, the more the thin-out value can be increased before the number of leaves in the average is severely reduced.

For a fixed number of points n in the data sample associated with the SRP, there is a trade-off between the computational efficiency (speed) of the sampling phase and the number of leaves in the final average histogram. The results discussed here suggest that the most computationally effective way to increase the number of leaves in the final average with a fixed number of chains is to increase the total number of transitions over which sampling takes place: taking the same number of samples with a higher thin-out value will give more leaves in the final average with relatively little computational cost, while simply increasing

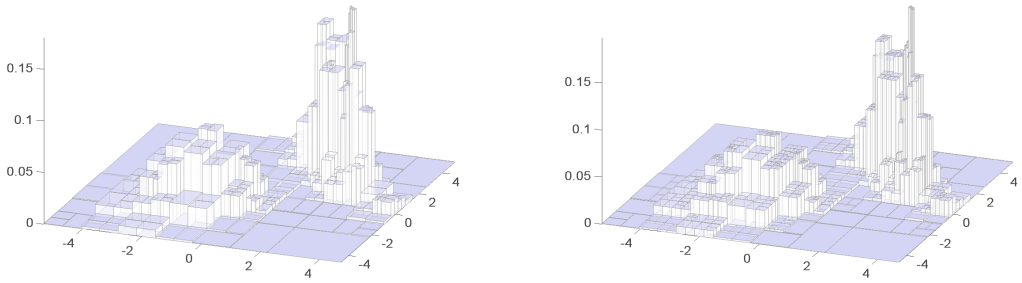
the number of SRP samples (for example, by decreasing the thin-out) will cost more and have relatively less effect on the number of leaves in the final average.

However, the tests described in Appendix H also found that, provided that $N > 1$, the effect of increasing the number of leaves in the final average RMRP density estimate on the L_1 error of that estimate against the true density was minimal. This reflects the influence that the natural Catalan prior has on the posterior distribution of SRP histograms for a finite n, the size of the data sample associated with the SRP. This has already been discussed in Section 6.5.3. The differences between the partitions of the SRP histograms sampled for a fixed n are small, relative to the total state space. Each minor difference in the partitions adds to the number of leaves in the final average without having much effect on the overall 'shape' of the average RMRP or the error against the true density. Importantly, however, the tests showed no evidence that, for a fixed data sample size n, the L_1 error *increased* with the number of leaves in the average.

Table 6.1 summarises the most important results discussed in Appendix H. The table shows the results for density estimates for data drawn from Example Density II (see Appendix B) using SRP samples taken from a single chain after convergence had been diagnosed using three chains and the $\widehat{R}$ convergence diagnostic as described in Section 6.7. The thin-out used for each set of results shown was 100 but the number of SRP samples N and the size n of the data sample were varied. The approximate L_1 errors and numbers of leaves shown in the table are averages over 10 replications of each SRP MCMC density estimation process. For each replication the L_1 error and number of leaves in the first SRP sampled (labelled '1^{st} sample' in the table) and in the final average RMRP ${}^{\square}\bar{f}$ itself were recorded. Table H.4 in Appendix H shows the same results in more detail.

Other tests showed that other individual SRP histogram states sampled also had L_1 errors similar to those shown for the first sampled state in Table 6.1. Comparing individual SRP errors and the error for the average RMRP ${}^{\square}\bar{f}$ shows that averaging reduced the estimation error. Comparing the results for the same data sample size $n = 50,000$ shows that increasing the number of SRP samples taken from the chain had no material affect on the error but more than doubled the number of leaves in the average. Comparing the results for $n = 50,000$ and $n = 100,000$ shows that increasing the sample size reduced the error and increased the number of leaves in the final average somewhat.

Figures 6.9 and 6.10 illustrate the effect on the average RMRP of increasing the number of sampling transitions compared with increasing the size of the data sample n. Figure 6.9 shows two RMRP estimates using a small sample of $n = 1,000$ data points drawn from example Density II, $d = 2$ (see Appendix B). Increasing both the thin-out value r and the number of SRP samples N gives a final average estimate with many more leaves but little actual additional smoothing. The data sample size is increased to $n = 10,000$ for Figure 6.10. Again, the most obvious effect of increasing the thin-out rate is to give a more complex version of the



(a) $r = 100$, $N = 100$, 265 leaves.

(b) $r = 10,000$, $N = 1,000$, 615 leaves.

Figure 6.9: Estimating Density II, $n = 1,000$.

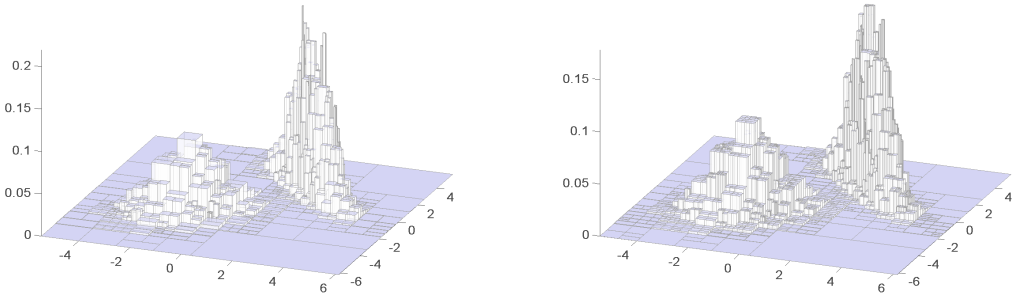
same basic shape of the density estimate, although comparing Figures 6.10(a) and 6.10(b) also indicates an improvement in the smoothness of the estimate around the higher mode in this particular example. The sample sizes used here are far smaller than are necessary to form a good estimate but the relatively low number of leaves in all the RMRPs shown makes it easier to distinguish the different effects of increasing the number of sampling transitions and increasing the data sample size.

The effect of increasing the number of chains from which samples are drawn on the number of leaves in the final average estimate and on the estimation error has not been thoroughly investigated in this thesis. Very limited tests (results not shown) suggest that the main effect of increasing the number of chains is to increase the time taken for the whole SRP MCMC process, because more chains require more calculations to compute the convergence diagnostic statistics. Doubling the number of chains had no effect on the approximated L_1 error in a small test using Example Density II ($d = 3$).

The results and examples shown in this thesis take thinned samples from all the chains used to diagnose convergence. Testing of the SRP MCMC method was carried out using thin-out values of 10 or 100 and averages over $N = 100$, 1,000 or 10,000 sampled SRP states. The results discussed in this section suggest that density estimates obtained using the larger values

Table 6.1: The influence of thin-out r, number of SRP samples N, and size of data sample n on average approximated L_1 error and average number of leaves, Density II ($d = 1$).

n	r	N	Average leaves		Average $\hat{L}_1$	
			1st sample	$\square\bar{f}$	1st sample	$\square\bar{f}$
50000	100	1000	69	591	0.042	0.037
50000	100	10000	76	1348	0.041	0.037
100000	100	1000	83	702	0.035	0.031



(a) $r = 100$, $N = 100$, 899 leaves.

(b) $r = 10,000$, $N = 1,000$, 2,662 leaves.

Figure 6.10: Estimating Density II, $n = 10,000$.

of N would have had a larger number of leaves in the final average but probably a similar error compared with equivalent estimates made using smaller N.

A number of tentative conclusions may be drawn from this investigation into sampling. The sample mean estimate RMRP can be expected (on the basis of the empirical evidence discussed here) to have a lower L_1 error than the individual SRP histograms sampled. Taking more SRP samples or spreading the SRP samples taken over a longer length of chain will increase the number of leaves in the average, and may improve some aspects of the estimate, but has little effect on the overall L_1 error. Most importantly, the L_1 error does not seem to increase with the number of leaves in the average for a fixed data sample size n. Finally, increasing the data sample size n decreases the estimation error.

6.7.3 Averaging

The final density estimate ${}^{\square}\bar{f}$ is obtained by converting each ${}^{\circ}s$ sampled to an RMRP using operation `SRPtoPCFDensity` (Algorithm 3.13) and averaging these RMRPs using Equation (3.12). This sample mean ${}^{\square}\bar{f}$ is an estimate of the Bayesian posterior expectation of the SRP histogram density estimate on a finite subspace of ${}^{\circ}\mathbb{S}_{0:\infty}$, the space of all possible SRPs with the same root box. Note that ${}^{\square}\bar{f}$ is an RMRP representation of a PCF and is not an RMRP representation of a *histogram* of the data: averaging RMRPs formed from SRP histograms does not give the same RMRP as an SRP histogram using the average *partition* over the SRP histograms samples.

This averaged histogram estimate may be contrasted to the average shifted histogram (ASH) (Scott 1992, chap. 5) and bootstrapped aggregate histogram (Klemelä 2009, chap. 17) already mentioned in Section 2.4. The ASH is not a data adaptive estimator and the 'sample' histograms over which the average is taken are wholly determined by the inputs to the algorithm. The bootstrapped aggregate histogram is an attempt to deal with the variability of a single data-adaptive histogram but each histogram averaged is not a random sample

from the sample space and, again, the average does not represent an estimate of the posterior expectation of the histogram density estimate.

6.7.4 Time taken to form a density estimate

The time taken to form an RMRP density estimate as the average of a number of SRP states sampled from the posterior distribution of such SRPs over a finite state space varies greatly with the size of the data sample n and the dimension d of the data, as well as with the sampling scheme used. Appendix H includes some timing results for selected densities in $d = 1$ and $d = 3$. The total time taken includes both the time taken to achieve convergence and the time taken to then collect the required number of samples from the chains. Both increase more than linearly with the sample size n because a larger n allows the chains to explore a larger space of more deeply split states. The characteristics of the underlying density also have an effect on the time, but in most cases dimensionality and data sample size seem to be the most important factors. There can also be large variations in timing between replications of the SRP MCMC process using different sequences of pseudo-random numbers.

The time taken for small data samples (of the order of $n = 10,000$) is very fast, a few minutes or less, depending on the number of SRP samples required. With a data sample size of $n = 50,000$ sampling 1,000 SRP states with a thin-out of 1,000 may again take seconds if $d = 1$, and minutes (but usually not hours) if $d = 3$, hours or days if $d = 5$. A much larger data sample size than $n = 50,000$ is required to be able to make an adequate density estimate for these higher dimensions. Tests with $n = 100,000$ for four and five-dimensional data drawn from a variety of different densities all took days. No tests were carried out for this thesis using the massive data sets discussed in (Sainudiin et al. 2013).

In summary, at around $d = 4$ or $d = 5$ the time required to make the density estimate with a large enough data sample size is very high, of the order of days or even weeks. It is much quicker in lower dimensions, often taking only seconds or minutes in $d = 1$, but the large data sample can be seen as a luxury, not a necessity, with one-dimensional data.

Very little effort has been made so far to create the fastest possible implementation of the SRP MCMC process, but some of the most obvious opportunities for efficiency have been utilised. The main issue is the very slow mixing of the chains — the inevitable result of the stay-split-merge base chain. Even if each transition takes place reasonably efficiently, the much larger effective state space that the prior allows the chains to explore when the data sample size is larger means that a huge number of transitions may be required to move an appreciable 'distance' within that state space. Even with the acceleration in convergence times achieved by selecting the initial states using the method described in Section 6.6.3, the number of transitions required to reach convergence increases with n and d and time taken by this pre-sampling phase increasingly dominates the overall process time (for a fixed thin-out and number of SRP samples taken).

The computational cost of finding the initial states also increases with n and d, but to a much lesser extent. This time is of the order of seconds for $n = 100,000$, even with $d = 100$. For $n = 1,000,000$ the time taken to find the starting states with $d = 100$ is about 20-30 minutes (a very small proportion of the total time that would then be required for the main MCMC process), and is only a few minutes for $d = 5$.

The convergence diagnostic method described in this chapter may be over-conservative, giving longer than necessary burn-in times. The efficiency of the calculations could probably also be improved. At present, however, the total time required to run the SRP MCMC process with large data sets (of the order of a million data points, or more) in $d = 4$ and $d = 5$ is a major drawback to the use of this method of forming a density estimate.

6.8 Alternative Monte Carlo Markov chain samplers

6.8.1 Independent Metropolis-Hastings

Section 6.5.1 discusses the mixing problems of the Metropolis-Hastings sampler with the stay-split-merge base chain. This sampler only allows the chain to move slowly, one split or merge at a time, around the state space, and is particularly vulnerable to becoming trapped in a small subset of states.

An alternative form of sampler is an independent Metropolis-Hastings sampler where the probability distribution of the proposals is independent of the current state. Appendix I discusses such a sampler and a possible implementation using a uniform proposal distribution. This independent Metropolis-Hastings sampler can, in theory, explore the state space more quickly than the stay-split-merge base chain sampler, and is not so vulnerable to becoming trapped. Convergence can also be assessed using coupled chains rather than independent chains (Johnson 1996). The vast number of possible SRP states even within a state space limited by restrictions such as a maximum number of leaves (see Section 6.3) will still cause mixing problems for the independent sampler, however, because the posterior mass is highly concentrated on small subsets of the state space and the rate of acceptance of the transition proposals will therefore be extremely low.

An implementation of the independent Metropolis-Hastings sampler with a uniform proposal distribution described in Appendix I was developed as part of the research for this thesis, to enable some broad comparisons with the stay-split-merge base chain sampler to be made. Very limited testing confirmed that acceptance rate is low but also suggested that the potential of such a sampler with more concentrated independent proposal might be worth pursuing further. Unfortunately, the higher the number of dimensions in the data and the larger the data sample associated with the SRP, the larger should be the finite state space allowed to the sampler and thus the independent sampler with a uniform proposal will struggle more on these larger problems, just as the stay-split-merge sampler does.

6.8.2 Other MCMC samplers

Many other forms of MCMC sampler could be tried. One possibility would be to utilise some of the results of the research for the independent sampler described above to develop a Metropolis-Hastings algorithm allowing a larger number of proposal states at each transition but keeping the proposals conditional on the current state to try to prevent the acceptance rate from becoming too low. For example, if the current state ${}^{\circ}s$ has $m = k + 1 \geq 2$ leaves (${}^{\circ}s \in {}^{\circ}\mathbb{S}_k$), then proposals could be drawn uniformly from all states in ${}^{\circ}\mathbb{S}_{k-1:k+1}$.

6.8.3 Perfect sampling

The coupling from the past (CFTP) algorithm developed by Propp and Wilson (1996) provides a way to obtain exact sample values from the stationary distribution of a Markov chain with a finite state space. However, if the state space is very large, CFTP is likely to be prohibitively computationally expensive (Lee 2008). Even a finite SRP state space, as described in Section 6.3, can be very large indeed. Monotone CFTP, also suggested by Propp and Wilson (1996), could therefore offer better prospects as a perfect sampler for SRPs. This form of CFTP requires a monotone Monte Carlo algorithm (Propp and Wilson 1996). A monotone Monte Carlo algorithm is one where a natural partial ordering $\preceq$ can be defined on the state space together with a minimal and maximal state and the partial ordering is preserved after a coupled transition (Lee 2008).

In terms of a MCMC sampler and a finite state space ${}^{\circ}\tilde{\mathbb{S}}$, this means that if two states ${}^{\circ}s^{(1)}, {}^{\circ}s^{(2)} \in {}^{\circ}\tilde{\mathbb{S}}$ satisfy the partial order with ${}^{\circ}s^{(1)} \preceq {}^{\circ}s^{(2)}$ and the same random vector R is used to propagate the transitions ${}^{\circ}s^{(1)} \to \varphi({}^{\circ}s^{(1)}, R)$, ${}^{\circ}s^{(2)} \to \varphi({}^{\circ}s^{(2)}, R)$ then $\varphi({}^{\circ}s^{(1)}, R) \preceq \varphi({}^{\circ}s^{(2)}, R)$ where $\varphi({}^{\circ}s, R)$ is the update function of the Markov chain (Lee 2008) such that

$$P\left(\varphi({}^{\circ}s, R) = {}^{\circ}s'\right) = P({}^{\circ}s, {}^{\circ}s') \text{ the probability of the transition } {}^{\circ}s \to {}^{\circ}s'.$$

Being able to use monotone CFTP to achieve perfect sampling depends on being able to identify a suitable natural partial ordering and update function to give a monotone MCMC algorithm. Given the difficulties encountered in establishing convergence using the MCMC sampler described above, perfect sampling using CFTP could provide a very useful improvement to the process for estimating the expectation of the posterior distribution of SRP histogram states. The possibilities for monotone CFTP for SRPs have been investigated briefly during this thesis but no appropriate algorithm has been identified. This remains an important area for future research.

6.9 Summary

Sainudiin et al. (2013) noted that a limitation of the posterior mean density estimate over

SRP histograms is the approximation of the likelihood of the data given an SRP $^{\circ}s$ by the maximum likelihood value from the histogram on $\mathbb{L}(^{\circ}s)$, the leaf boxes of $^{\circ}s$. Subject to this, Teng (2013) and Sainudiin et al. (2013) showed that the Metropolis-Hastings MCMC process described in Section 6.4 has some potential as a method for making density estimates for high dimensional unstructured data. One of the objectives of this thesis is to extend the basic Metropolis-Hastings MCMC method (Sainudiin et al. 2013) as a density estimator for more structured (non-uniform) densities. The semi-automatic SRP MCMC described in this chapter has partially succeeded in achieving this objective.

Assessing convergence is one of the most difficult aspects of the MCMC process. The following conclusions are drawn from the investigations discussed in the Sections 6.6 and 6.7:

- Convergence must be assessed using multiple chains and more than one low-dimensional scalar summary of the SRP state.
- Some form of automated assessment of the convergence and initiation of sampling is desirable because of the length of time typically taken to run an SRP MCMC process.
- Supplementary information, such as trace plots of the scalar values used to assess convergence and the components of the calculation of the convergence diagnostic(s) used, must also be examined after the process has ended to confirm that the automated sampling method chose a suitable burn-in time.

Further research on SRP MCMC convergence could investigate more useful scalar summaries of SRP histogram state, the method of calculating the interval-based $\widehat{R}$ convergence diagnostic described here and a comparison with the results of using $\widehat{R}_{\text{variance}}$ instead, appropriate tolerance values for the convergence diagnostic, and the benefits of using more chains to diagnose convergence. Other methods for finding initial states from which to start the MCMC process could also be developed.

The natural Catalan prior used in the current implementation of the SRP MCMC process has a strong influence on the final density estimate. In general, using a larger data sample will allow the chains to explore more deeply split states. The current implementation may be greatly improved by the development of alternatives to the natural Catalan prior that will allow the chains to explore more deeply split states for the same data sample size.

No theoretical work has been done on the properties of the final average RMRP density estimate. The limited empirical results described in this chapter suggest that, for the densities and sample sizes tested, the average RMRP obtained from the SRP MCMC process provides an L_1-consistent estimator. As the size of the data sample increases the L_1 error falls, irrespective of the complexity of the average density estimate itself (which is influenced by the number of SRP samples averaged and the thin-out value used). Much more empirical research is required to strengthen this tentative conclusion, even if a full theoretical treatment of the subject is not possible.

One of the main practical drawbacks of the SRP MCMC process implemented for this thesis is that the total time required to run the SRP MCMC process with large data sets (of the order of a million data points, or more) in $d = 4$ and $d = 5$ is of the order of days or weeks.

Some of the mixing problems experienced with the Metropolis-Hastings sampler with a stay-split-merge base chain might be alleviated by allowing a larger range of possible proposal states at each transition. A fully independent Metropolis-Hastings sampler with a uniform proposal is likely to experience prohibitively low acceptance rates but further research should be carried out into a less restricted form of conditional proposal distribution, or a concentrated (non-uniform) independent proposal distribution.

Assessing convergence heuristically will be a major issue using any of these samplers. Perfect sampling using a monotone MCMC algorithm would avoid this difficulty but no progress has yet been made in developing such an algorithm.

Chapter 7

Regular paving approximation of a kernel density estimate

7.1 Introduction

The RMRP density estimate discussed in Chapter 6 is the sample mean estimate of the Bayesian posterior expectation of the SRP histogram of the sample data set.

Section 2.5 in Chapter 2 describes multivariate kernel density estimators and the properties of a kernel density estimate (KDE). The main theoretical advantages of a KDE compared with a histogram are that it gives a smooth (differentiable) estimate with faster convergence to the true density (Scott and Sain 2005). The advantages of the RMRP representation relate to the computational efficiency of the operations that can then be carried out on the density estimate, including obtaining pointwise densities, conditional density estimates, marginal estimates, coverage regions, etc., as described in Section 3.6.2.

One way to combine some of the advantages of both an RMRP and a KDE would be to approximate a KDE using a PCF represented as an RMRP. This chapter discusses how such an approximation could be obtained and gives some examples comparing KDEs, averaged histogram RMRP estimates, and RMRP approximations to the KDEs.

7.2 Obtaining the kernel density estimate

The methods discussed below in Section 7.3 to approximate a KDE using a PCF represented as an RMRP require only that the KDE $\hat{f}_K$ can be evaluated pointwise at any point x of interest. The approximation method can then be used to approximate any KDE and the problem of how to create the KDE can be regarded as independent of the problem of how to create the RMRP approximation to the KDE. To test the approximation method and to make some preliminary comparisons between KDEs, RMRP approximations of KDEs, and RMRPs formed using the SRP MCMC method of Chapter 6, it was necessary to implement at least one kernel density estimator suitable for multivariate data. The KDEs in this chapter are created using the MCMC method for finding optimal bandwidths described in Zhang et al. (2006). `C` code for this algorithm was kindly supplied by Dr. Xibin Zhang and Professor Maxwell L. King of Monash University and converted to `C++` for use in the research for this thesis.

7.3 Approximating a kernel density estimate with a real-mapped regular paving

A KDE $\hat{f}_K$ can be approximated as a PCF represented as an RMRP ${}^{\square}f$ by mapping each node $\rho\mathsf{v}$ in ${}^{\square}f$ to a value in $\mathbb{R}$ that is calculated by reference to $\hat{f}_K$ and its box $\boldsymbol{x}_{\rho\mathsf{v}}$. Let $\texttt{GetValue}(\hat{f}_K, \rho\mathsf{v})$ be some procedure for calculating $f_{\rho\mathsf{v}} \in \mathbb{R}$ for node $\rho\mathsf{v}$ given a KDE $\hat{f}_K$. The approximation can be obtained by an adaptation of the RPQ function approximation algorithm (Algorithm 6, $\texttt{RPQEnclose}^{\triangledown}$) in Harlow et al. (2012). Algorithm 7.1 gives an RPQ procedure for approximating a KDE with an RMRP ${}^{\square}f$. The final step is to normalise the RMRP.

Algorithm 7.1: $\texttt{RPQApproximate}({}^{\square}f, \hat{f}_K, \psi, \overline{\psi}, \overline{m})$

input : RMRP ${}^{\square}f$ with root node ρ and root box $\boldsymbol{x}_\rho$,
priority function $\psi : \mathbb{L}^{\triangledown}({}^{\square}) \to \mathbb{R}$,
$\overline{\psi}$ the maximum value of $\psi(\rho\mathsf{v}) \in \mathbb{L}^{\triangledown}({}^{\square}f)$ in the final RMRP,
$\overline{m}$ the maximum number of leaves in the final RMRP.

output : ${}^{\square}f$ such that $\mathbb{L}^{\triangledown}({}^{\square}f) = \emptyset$ or $\psi(\rho\mathsf{v}) \leq \overline{\psi}\ \forall \rho\mathsf{v} \in \mathbb{L}^{\triangledown}({}^{\square}f)$ or $|\mathbb{L}({}^{\square}f)| \leq \overline{m}$.

while $\mathbb{L}^{\triangledown}({}^{\square}f) \neq \emptyset\ \ \&\ \ |\mathbb{L}({}^{\square}f)| < \overline{m}\ \ \&\ \ \psi\left(\text{argmax}_{\rho\mathsf{v} \in \mathbb{L}^{\triangledown}({}^{\square}f)}\, \psi(\rho\mathsf{v})\right) > \overline{\psi}$ **do**

- $\rho\mathsf{v} \leftarrow \texttt{random\_sample}\left(\underset{\rho\mathsf{v} \in \mathbb{L}^{\triangledown}({}^{\square}f)}{\text{argmax}}\ \psi(\rho\mathsf{v})\right)$
- Split $\rho\mathsf{v}$: $\triangledown_{\text{RP}}(\rho\mathsf{v}) = \{\rho\mathsf{vL}, \rho\mathsf{vR}\}$ `// split the sampled node`
- $\boldsymbol{f}_{\rho\mathsf{vL}} \leftarrow \texttt{GetValue}(\hat{f}_K, \rho\mathsf{vL})$
- $\boldsymbol{f}_{\rho\mathsf{vR}} \leftarrow \texttt{GetValue}(\hat{f}_K, \rho\mathsf{vR})$
- $\texttt{RPQApproximate}({}^{\square}f, \hat{f}_K, \psi, \overline{\psi}, \overline{m})$

end

$\texttt{Normalise}({}^{\square}f)$

A variety of different priority functions ψ could be used. The priority function used to obtain the results discussed in this chapter is based on the total variation (Devroye and Lugosi 2001, chap. 5), evaluated over the box $\boldsymbol{x}_{\rho\mathsf{v}}$ of a leaf node $\rho\mathsf{v}$, between the estimate given by $\boldsymbol{f}_{\rho\mathsf{v}}$ and the estimate given by $\boldsymbol{f}_{\rho\mathsf{vL}}$ and $\boldsymbol{f}_{\rho\mathsf{vR}}$ where $\rho\mathsf{vL}$, $\rho\mathsf{vR}$ are respectively the left and right child nodes of $\rho\mathsf{v}$ that would be formed if $\rho\mathsf{v}$ were to be split. $\psi(\rho\mathsf{v})$ is thus a measure of the change in the function estimate if $\rho\mathsf{v}$ is split. Leaf nodes with the largest potential change impact on the function estimate, as defined by this ψ, are split first.

$$\psi(\rho\mathsf{v}) = \frac{1}{2}\text{vol}\left(\boldsymbol{x}_{\rho\mathsf{v}}\right)\left(|f_{\rho\mathsf{v}} - \boldsymbol{f}_{\rho\mathsf{vL}}| + |f_{\rho\mathsf{v}} - \boldsymbol{f}_{\rho\mathsf{vR}}|\right) \tag{7.1}$$

Similarly, a variety of different specifications of $\texttt{GetValue}(\hat{f}_K, \rho\mathsf{v})$ could be used provided that the value $\boldsymbol{f}_{\rho\mathsf{v}}$ mapped to a node $\rho\mathsf{v}$ is in some way determined by an evaluation of $\hat{f}_K$ in relation to the box $\boldsymbol{x}_{\rho\mathsf{v}}$ associated with the node $\rho\mathsf{v}$. The method used to obtain the results

discussed in this chapter is very simple:

$$\texttt{GetValue}(\hat{f}_K, \rho\mathsf{v}) := \hat{f}_K\left(\text{mid}\left(\square(\boldsymbol{x}_{\rho\mathsf{v}})\right)\right) \quad ,$$

where $\square(\boldsymbol{x}_{\rho\mathsf{v}})$ is the interval hull of the box $\boldsymbol{x}_{\rho\mathsf{v}}$. Thus the value mapped to a node $\rho\mathsf{v}$ is the KDE density evaluated at the mid-point of the interval hull of the box $\boldsymbol{x}_{\rho\mathsf{v}}$ associated with the $\rho\mathsf{v}$. A more computationally intensive method would be to use the average pointwise density evaluated at a number of random or quasi-random (Niederreiter 1992) points in $\square(\boldsymbol{x}_{\rho\mathsf{v}})$. Both of these forms of `GetValue` only use the KDE evaluated at individual points. This means that it is not possible to enclose the target function rigorously and produce an approximation with a guaranteed maximum error.

In contrast, Algorithm 6, $\texttt{RPQEnclose}^{\triangledown}$, in Harlow et al. (2012) uses an inclusion function $\boldsymbol{g}$ of the target function g to give an $\mathbb{IR}$-MRP (interval-mapped regular paving) representing a rigorous interval enclosure of g over the root box of the $\mathbb{IR}$-MRP. If the interval-mapped approximation is turned into an RMRP approximation by setting $f_{\rho\mathsf{v}} \leftarrow g(\text{mid}\left(\square(\boldsymbol{x}_{\rho\mathsf{v}})\right))$ for each node $\rho\mathsf{v}$ in the tree, then the maximum error between $\boldsymbol{g}(x)$ and the pointwise image of x under the RMRP approximation for any $x \in \boldsymbol{x}_\rho$ is bounded above by the diameter of the widest interval $\boldsymbol{f}_{\rho\mathsf{v}}$ mapped to any leaf node $\rho\mathsf{v}$ in the $\mathbb{IR}$-MRP.

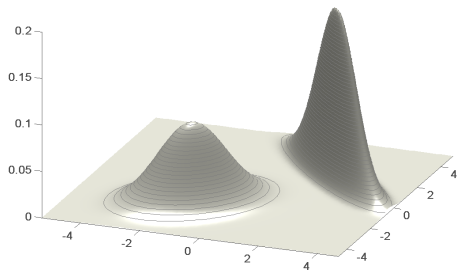
It might be possible to calculate an inclusion function of the KDE by summing over the inclusion functions of each kernel component. In that case, it would be possible to rigorously enclose a target KDE $\hat{f}_K$ and produce an approximation with a guaranteed maximum error against $\hat{f}_K$. A potential disadvantage of this approach is that the final inclusion function of the KDE produced by summing over a relatively large number of components could give such wide interval enclosures that a very large number of leaves (fine partition) would be required to achieve a reasonable error bound. The potential for using a KDE inclusion function in the approximation is not explored further in this thesis but would be an interesting avenue for future investigation.

The RPQ operation allows two forms of 'stopping condition' to be specified. These are similar to the stopping conditions discussed in Chapter 5 and Appendix D: an overall maximum number of leaves $\overline{m}$ in the RMRP; and stopping condition $\overline{\psi}$ related to the priority function itself. Splitting ceases when the largest 'value' under ψ of any splittable leaf node is less than or equal $\overline{\psi}$, i.e., when $\max\{\psi(\rho\mathsf{v}) : \rho\mathsf{v} \in \mathbb{L}^{\triangledown}({}^{\square}f)\} \leq \overline{\psi}$, or when the number of leaves in the RMRP is $\overline{m}$, or when there are no more splittable leaf nodes in the RMRP (see Section C.3 in Appendix C for implementation-imposed limits on splittable RP leaf nodes).

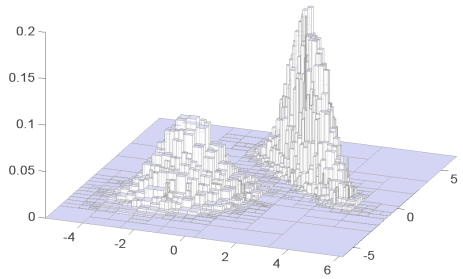
7.4 Bivariate density example

This example uses the 2-dimensional version of Density II described in Appendix B. This bivariate density is the same as Density A studied in Zhang et al. (2006).

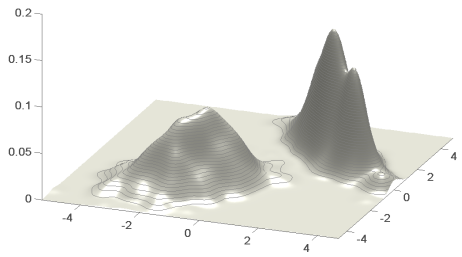
Figure 7.1 shows estimates of Density II using $n_K = 1,000$ sample points simulated from the true density. Figure 7.1(a) is a visualisation[1] of the true density. Figure 7.1(b) is the RMRP formed using $n = 50,000$ data points drawn from Density II and averaging 100 SRP histogram samples taken after burn-in from a Markov chain (with thin-out 100), using the method described in Chapter 6. Figure 7.1(c) is a visualisation of the KDE created using the optimal diagonal bandwidth matrix as described in Zhang et al. (2006) (with the same burn-in of 5,000 iterations and the same 250,000 recorded iterations as were used in that study). Figure 7.1(d) is a visualisation of the KDE created using the 'Normal reference rule'.



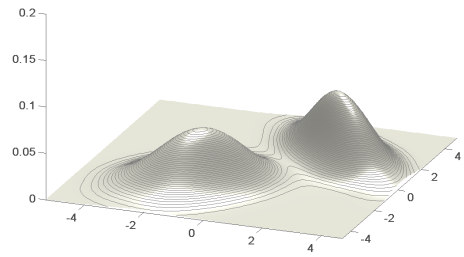
(a) True density.



(b) Averaged histogram RMRP ($n = 50,000$, 1,551 leaves).



(c) MCMC bandwidth KDE.



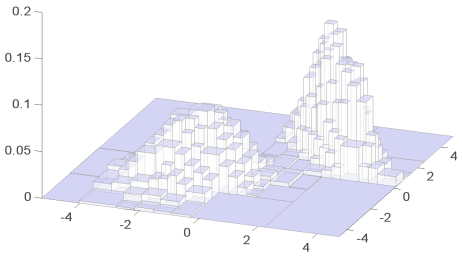
(d) Normal reference rule KDE.

Figure 7.1: Estimating Density II ($n_K = 1000$).

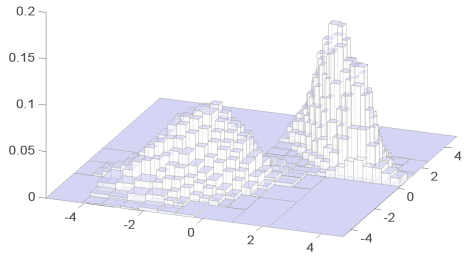
Figure 7.1 illustrates the smoothness of the KDE method over a histogram-based method but also demonstrates the oversmoothing that can occur if the KDE bandwidth matrix is not suitable, such as bandwidths chosen using the Normal reference rule used with non-Normal data (Figure 7.1(d)).

[1]This figure and similar density visualisations shown in this thesis are created using the MATLAB® `surf` function with a grid of points in the required domain and densities evaluated at each grid-point.

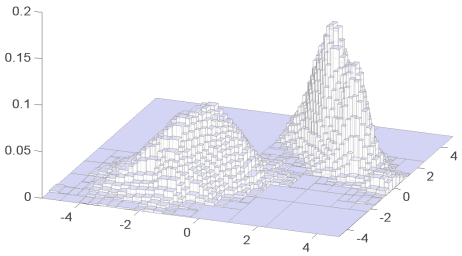
Figure 7.2 shows the results of using `RPQApproximate` to approximate the KDE shown in Figure 7.1(c) with `GetValue`$(\hat{f}_K, \rho\mathsf{v}) := \hat{f}_K\left(\mathrm{mid}\left(\square(\boldsymbol{x}\rho\mathsf{v})\right)\right)$ as described above and priority function ψ as in Equation (7.1) for various values of $\overline{\psi}$.



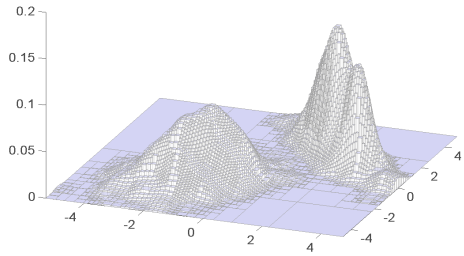
(a) $\overline{\psi} = 0.001$ (187 leaves).



(b) $\overline{\psi} = 0.005$ (316 leaves).



(c) $\overline{\psi} = 0.0001$ (919 leaves).

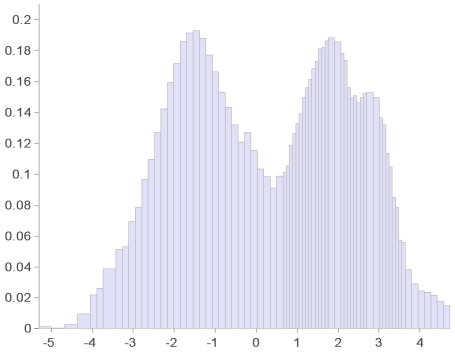


(d) $\overline{\psi} = 0.00001$ (4420 leaves).

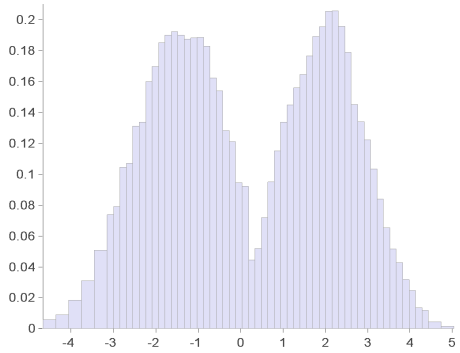
Figure 7.2: Approximating the KDE ($n_K = 1,000$) using an RMRP.

The time taken to make these approximations increases as $\overline{\psi}$ decreases but is of the order of seconds (compared with minutes, or hours, or days, depending on n_K, for the KDE itself).

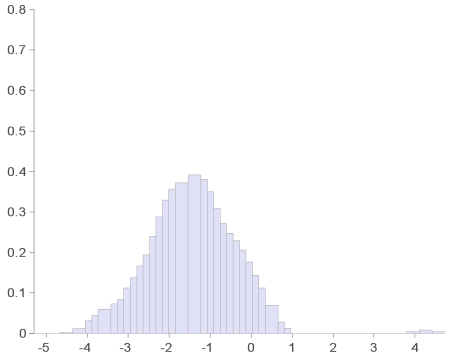
Figure 7.3 shows some of the various operations that can then be carried out on the RMRP approximation of the KDE. The RMRP used here is that shown in Figure 7.2(a) ($\overline{\psi} = 0.0001$). Figures 7.3(a) and 7.3(b) show the marginalised approximations on coordinates 1 ($^{\square}f^{\{1\}}(x_1)$) and 2 ($^{\square}f^{\{2\}}(x_2)$), respectively. Figure 7.3(c) shows $^{\square}f^{|x_2=-1.5}(x_1)$, the normalised slice on $x_2 = -1.5$, an estimate of $f_{II}(x_1 \,|\, x_2 = -1.5)$ the univariate conditional density of x_1 given $x_2 = -1.5$. Figure 7.3(d) shows $^{\square}f^{|x_1=2.0}(x_2)$, the normalised slice on $x_1 = 2.0$, an estimate of $f_{II}(x_2 \,|\, x_1 = 2.0)$ the univariate conditional density of x_2 given $x_1 = 2.0$. Figure 7.3(e) shows the 95% (dark-gray) and 80%(mid-gray) coverage regions of the RMRP against the 100% coverage (light-gray) whole.



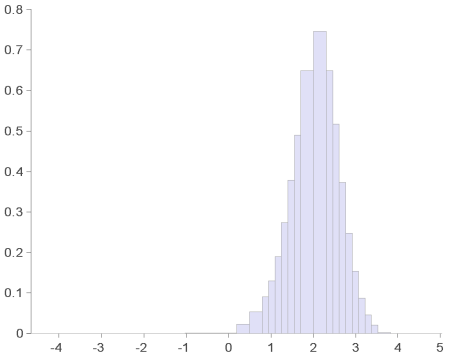
(a) Marginal on coordinate 1, ${}^{\square}f^{\{1\}}(x_1)$.



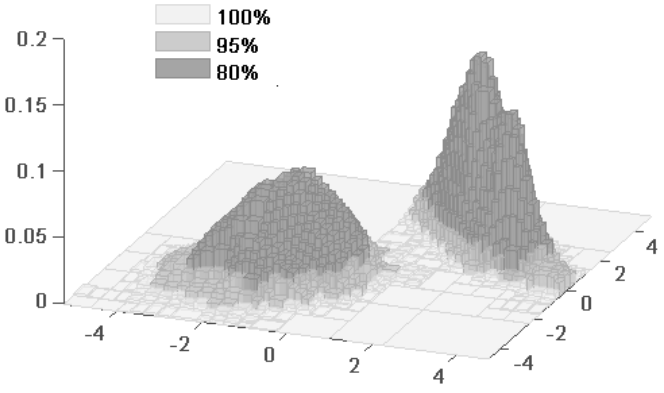
(b) Marginal on coordinate 2, ${}^{\square}f^{\{2\}}(x_2)$.



(c) Conditional density ${}^{\square}f(x_1 \,|\, x_2 = -1.5)$.



(d) Conditional density ${}^{\square}f(x_2 \,|\, x_1 = 2.0)$.



(e) Coverage regions of ${}^{\square}f(x_1, x_2)$.

Figure 7.3: Density operations using the RMRP approximation with $\overline{\psi} = 0.0001$.

7.5 How close should the approximation be?

An RPQ algorithm will continue to split the RMRP (refine the partition of the root box) until some stopping condition is satisfied. The $\overline{\psi}$ condition indicates when the approximation is considered to be close enough to the target; the $\overline{m}$ condition is used to ensure that splitting will eventually stop at some point (dictated perhaps by available memory) even if the $\overline{\psi}$ condition is not satisfied. Even if a non-RPQ-based algorithm was used to approximate the KDE, some way of determining how close the approximation should try to be would be necessary.

This section discusses the effect of different levels of $\overline{\psi}$ on the accuracy of the approximation with $\texttt{GetValue}(\hat{f}_K, \rho\mathsf{v}) := \hat{f}_K\left(\text{mid}\left(\square(\boldsymbol{x}\rho\mathsf{v})\right)\right)$ and priority function ψ as in Equation (7.1).

When the true distribution of the data is unknown it would only be possible to measure the error of the approximation against the KDE, but for the purpose of investigating the merits of the approximation method in general it is also interesting to compare measures of the accuracy of both the KDE and the approximation of the KDE to the true density.

Zhang et al. (2006) used the Kullback-Leibler loss as a measure of the discrepancy between a density estimate $\hat{f}$ and the true distribution f and estimated this by simulating a large number N of random points from the true density and using this to calculate

$$\hat{d}_{KL}\left(f, \hat{f}\right) = \frac{1}{N}\sum_{i=1}^{N} \log\left(\frac{f(x_i)}{\hat{f}(x_i)}\right). \tag{7.2}$$

Hall (1987) gives an interesting discussion of the Kullback-Leibler loss in relation to the choice of KDE bandwidths using likelihood cross-validation, noting the importance of "the interaction between the tail properties of the kernel K and of the unknown density f" (Hall 1987, p. 1492) and that minimising the Kullback-Leibler loss can give bandwidths that oversmooth in L_2 error terms. To facilitate comparison with the results of Zhang et al. (2006) the comparisons shown in this section include estimates of the Kullback-Leibler loss calculated using Equation (7.2) and $N = 1,000,000$. In addition, an estimate of the L_1 error is shown. This was calculated using quasi-Monte Carlo integration (Niederreiter 1992) over $N = 1,000,000$ quasi-random points to estimate $\int|f - \hat{f}|$ as

$$\hat{L}_1\left(f, \hat{f}\right) = \frac{\text{vol}\left(\boldsymbol{x}_\rho\right)}{N}\sum_{i=1}^{N}|f(x_i) - \hat{f}(x_i)| \tag{7.3}$$

using Equation (3.6.2.7). Both the Kullback-Leibler loss and the L_1 error are 'dimensionless' in the sense that these measures can be compared for densities in different dimensions, but the fixed number of error estimation points used for the results shown here will be increasingly sparsely distributed as the dimensions increase.

The true densities used for the results given in this section are the $d = 2, 3, 4$ and 5 versions of the multivariate Example Density II given in Appendix B.

The results for $\hat{d}_{KL}$ and $\hat{L}_1$ for true multivariate density Density II and the estimate $\hat{f}$ as, first, the KDE (with $n_K = 2,000$), and then the RMRP approximation to the KDE with $\overline{\psi} = 0.000001$ for $d = 2, 3, 4$ and 5, are shown in Table 7.1. The time taken for each estimate and the numbers of leaves in the RMRP approximations are also shown. Estimated errors are given to two decimal places, KDE times are rounded to indicate the general magnitude of times from several different replications, and the RMRP approximation times are given to one decimal place. Other than the KDE times, these results are from just one replication of the process. The values of $\hat{d}_{KL}$ and $\hat{L}_1$ and the number of leaves in the RMRP did not vary greatly over a limited number of repetitions with different data sets and different sequences of pseudo-random numbers in the RPQ process (results not shown). The timings were recorded when other processes running on the same machine and can only be taken as a general indication of the time required to make these estimates.

Table 7.1: Estimated errors for KDE ($n_K = 2,000$) and RMRP-KDE approximation ($\overline{\psi} = 0.000001$).

	$\hat{d}_{KL}$	$\hat{L}_1$ error	Time (s)	Leaves
2-d				
KDE	0.04	0.20	5,000–7,200	n/a
RMRP-KDE approximation	0.15	0.20	10.2	21,105
3-d				
KDE	0.13	0.35	5,600–7,200	n/a
RMRP-KDE approximation	1.72	0.40	23.7	49,480
4-d				
KDE	0.25	0.51	7,200–8,050	n/a
RMRP-KDE approximation	4.11	0.59	78.3	90,621
5-d				
KDE	0.41	0.66	7,350–8,880	n/a
RMRP-KDE approximation	3.33	0.76	66.1	133,493

Full results for a range of values of $\overline{\psi}$ are shown in Tables J.1, J.2, J.3, and J.4 in Appendix J. The estimated Kullback-Leibler loss for the 2-d KDE (0.04) accords well with the figure given in Zhang et al. (2006) (0.058 for $n_K = 1,000$). The Kullback-Leibler loss estimates for the RMRP approximations to the KDE are very large. It appears that these may be particularly affected by the specification of `GetValue` used here. Using only the mid-point of a box to get a value for the RMRP approximation over the whole box may give a particularly 'loose' estimate in the tails, especially as the number of dimensions increases.

The full results, using other values of $\overline{\psi}$, shown in Appendix J suggest that the improvements in the RMRP approximation (as measured by the estimated L_1 error compared with the estimated L_1 error for the KDE) achieved from further reductions in $\overline{\psi}$ after about $\overline{\psi} = 0.000001$ fall off rapidly. The number of leaves in the resulting RMRP approximation increases approx-

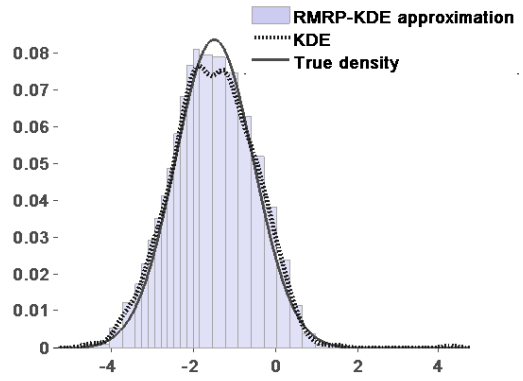
imately in inverse proportion to the change in $\overline{\psi}$ but the effect on the estimated error is increasingly small. Tests using other densities should be carried out to confirm that this effect is not particular to Density II.

The KDE and the RMRP approximation to the KDE can be compared with the true density by taking 1-dimensional slices through the approximation and overlaying a representation of the corresponding 1-dimensional profile of the KDE and true density, as in Figure 7.4. The slice from the RMRP approximation to the KDE is taken using `MRPSlice` operation (see Algorithm 3.7, Section 3.5.2). Figure 7.4(a) shows a slice parallel to coordinate 1 of the RMRP approximation to the KDE using the bivariate version of Density II. The slice is taken at $x_2 = -1.5$ and thus shows an approximation to an *unnormalised* conditional function $f_{II}(x_1 \,|\, x_2 = -1.5)$. The profile of the true density is drawn using the bivariate f_{II} evaluated at a regular sequence of points $\{(x_1, -1.5)\}$ across the support of the sample data on coordinate 1. Similarly the profile of the KDE is drawn using $\hat{f}_K$ evaluated at the same sequence of points $\{(x_1, -1.5)\}$.

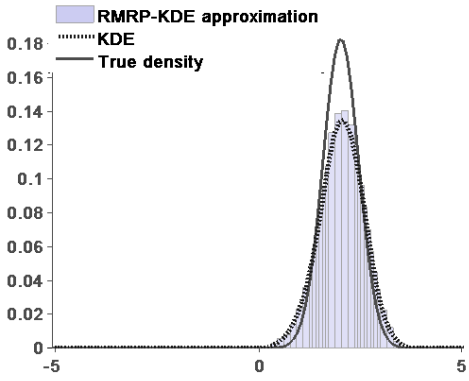
Figure 7.4(b) shows a slice parallel to coordinate 2 of the RMRP approximation with $\overline{\psi} = 0.0001$ to the KDE taken at $x_1 = 2.0$ (an approximation to an *unnormalised* conditional function $f_{II}(x_2 \,|\, x_1 = 2.0)$. The profiles of the true density and the $\hat{f}_K$ are drawn using f_{II} and $\hat{f}_K$ evaluated at a regular sequence of points $\{(2.0, x_2)\}$ across the support of the sample data on coordinate 2. Figures 7.4(c) and Figure 7.4(d) show equivalent 1-dimensional slices and profiles using the $d = 3$ version of f_{II} as the true density, a KDE of sample data from the true density and an RMRP approximation to the KDE ($\overline{\psi} = 0.00001$). Figures 7.4(e) and Figure 7.4(f) show equivalent 1-dimensional slices and profiles using the $d = 5$ true density, a KDE, and an (RMRP approximation to the KDE with $\overline{\psi} = 0.000001$). The effect of the very loose approximation given by the mid-point `GetValue` is particularly apparent as the number of dimensions increases.

The slices and profiles in Figure 7.4 are chosen to run through the two modes of the true density, Density II. Figure 7.4 illustrates the increased tendency to oversmoothing at the modes as the number of dimensions increases that will be experienced when using a diagonal bandwidth matrix and which is described in Section 7.2 (the KDE shown in Figures 7.4(e) and Figure 7.4(f) may also be affected by the relatively small sample size for this number of dimensions). The RMRP approximation to the KDE will naturally show similar oversmoothing to that in the KDE.

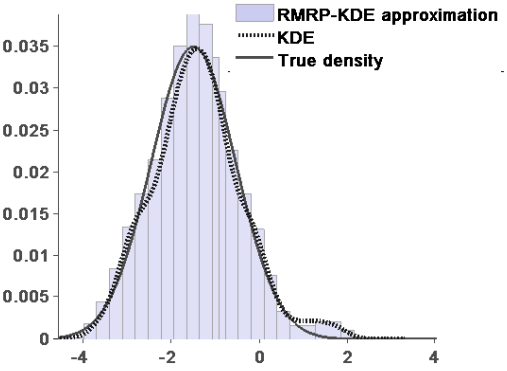
The main conclusions from these results, which are based on only one test density, are that the RMRP approximations can be made very quickly, at least for the range of $\overline{\psi}$ considered here and using an easily-evaluated specification of `GetValue`, and that initial decreases in $\overline{\psi}$ give the greatest improvements in the estimated error and incremental improvements in the error for further decreases in $\overline{\psi}$ tail off rapidly although the complexity (number of leaves) of the RMRP approximation continues to increase.



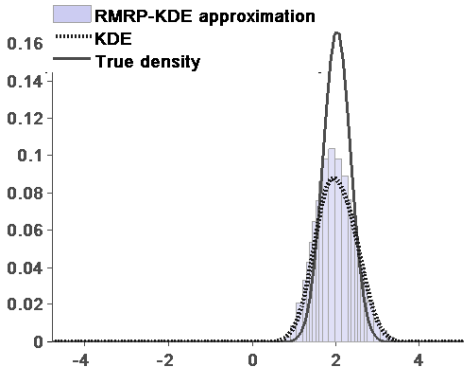
(a) $d = 2$, $x_2 = -1.5$, $\overline{\psi} = 0.0001$.



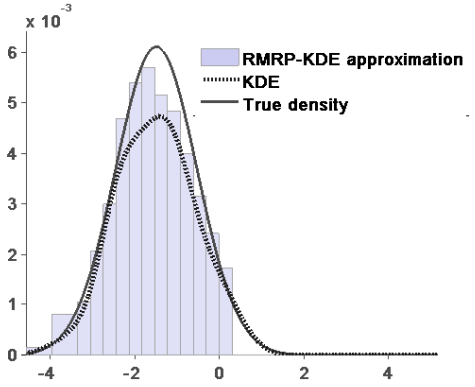
(b) $d = 2$, $x_1 = 2.0$, $\overline{\psi} = 0.0001$.



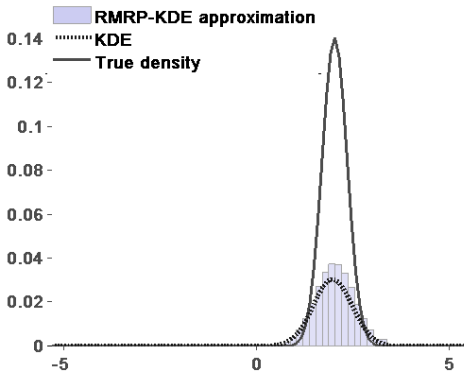
(c) $d = 3$, $x_2 = x_3 = -1.5$, $\overline{\psi} = 0.00001$.



(d) $d = 3$, $x_1 = x_3 = 2.0$, $\overline{\psi} = 0.00001$.



(e) $d = 5$, $x_2 = x_3 = x_4 = x_5 = -1.5$, $\overline{\psi} = 0.000001$



(f) $d = 5$, $x_1 = x_3 = x_4 = x_5 = 2.0$, $\overline{\psi} = 0.000001$.

Figure 7.4: Density II, KDE and RMRP-KDE approximation slice.

These results also suggest that the mid-point `GetValue` may be too inaccurate; alternatives should be investigated and improvements in accuracy compared with an increase in computational complexity/computation time taken. Similar error measurements should also be made using other known densities.

It is also important to recall the discussion above relating to the form of `GetValue`. Using only pointwise evaluations of the KDE can be computationally efficient but it is then not possible to guarantee how 'close' the approximation ${}^{\square}f$ is to the KDE. In particular, say that, when the algorithm terminates, ${}^{\square}f$ has k splits (i.e., ${}^{\square}f \in {}^{\square}\mathcal{F}_k$) and $\psi(\rho\mathsf{v}) \leq \overline{\psi}$, $\forall \rho\mathsf{v} \in \mathbb{L}^{\triangledown}({}^{\square}f)$. There could still be some ${}^{\square}f' \in {}^{\square}\mathcal{F}_{(k+1):\infty}$ with the same root box (i.e., a more split version of ${}^{\square}f$) where $\psi(\rho\mathsf{v}') > \overline{\psi}$ for some $\rho\mathsf{v}' \in \mathbb{V}({}^{\square}f')$ and there could be important features in the KDE that are not be reflected in the approximation ${}^{\square}f$.

One of the issues to be considered concerning the `RPQApproximate` procedure is how to specify suitable values for the stopping conditions $\overline{\psi}$ and $\overline{m}$ when the true density is unknown. As with SRPs and SEB-based RPQs, it could be extremely hard to identify suitable values for these parameters.

7.6 Performance evaluation

An optimal KDE is certainly superior to an optimal histogram or averaged RMRP histogram, but given that, in practice, it is hard to achieve the promise of the optimal KDE it is interesting to compare some sub-optimal but feasible KDEs with some sub-optimal but feasible RMRP density estimates obtained using the averaged MCMC SRP histogram method described in Chapter 6.

Table 7.2 shows a summary of results for $\hat{d}_{KL}$ and $\hat{L}_1$ with true density Density II (see Appendix B) for $d = 2, 3, 4$ and 5. The estimates of the Kullback-Leibler loss and the L_1 error, $\hat{d}_{KL}$ and $\hat{L}_1$, respectively, are shown for the KDE (using $n_K = 2,000$ sample points from the true density) and also for an RMRP estimate formed by averaging 100 SRP histogram samples taken after burn-in from a Markov chain (thin-out 100), using the method described in Chapter 6. The effect on $\hat{d}_{KL}$ and $\hat{L}_1$ of increasing the data sample size n used for the averaged histogram estimate is shown in the table. The number of leaves in the final averaged RMRP is also shown. The results discussed in Section 6.7.2 and Appendix H suggest that using a higher thin-out value or more SRP samples would have increased the number of leaves in the averaged histogram estimate (and the time taken to make the estimate), but that the estimated errors would have been very close to those shown here.

The RMRP estimates realised by averaging RMRP histogram samples from a Markov chain will differ with different data samples and also with the sequence of pseudo-random numbers used to determine proposal and acceptance for each transition in the chain. The values shown for $\hat{d}_{KL}$ and $\hat{L}_1$ for the averaged histogram estimate in Table 7.2 are averages over 10 replications of the process with different sample data and different pseudo-random number

sequences used for each replication, except for the results for $d = 5$, $n = 100,000$, where time constraints meant that only three replications could be completed. The variability between replications is low (for example, for $d = 2$ and $n = 10,000$, minimum $\hat{d}_{KL} = 0.058$, maximum 0.067; minimum $\hat{L}_1 = 0.217$, maximum 0.228) and the averages give a reasonable summary of the estimated errors calculated. There is wider variability in the time taken for each replication and the number of leaves in the final averaged RMRP and so the minimum and maximum over the 10 replications (three replications for $d = 5$, $n = 100,000$) are shown for both time taken and leaves for the averaged histogram RMRPs. The averaged histogram estimate timings were recorded when other processes running on the same machine and can only be taken as a general indication of the time required to make these estimates.

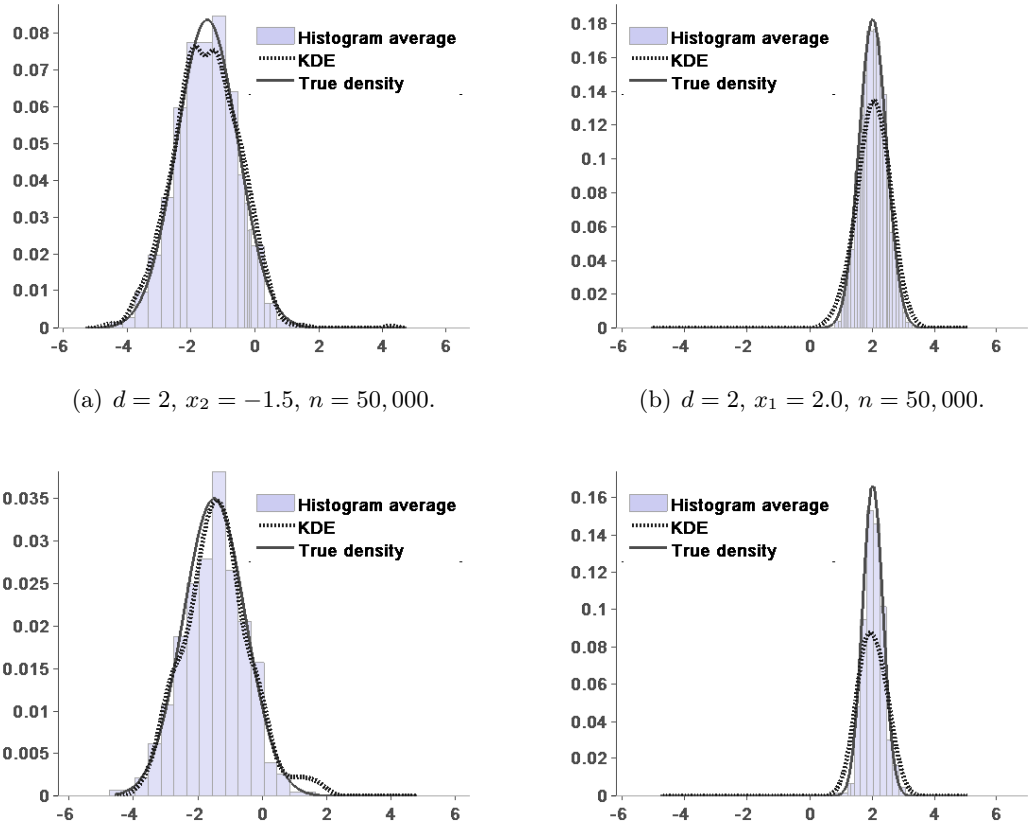
Table 7.2: Estimated errors for KDE and averaged SRP histogram RMRP.

	$\hat{d}_{KL}$	$\hat{L}_1$ error	Time (s)		Leaves	
			min.	max.	min.	max.
2-d						
KDE ($n_K = 2,000$)	0.04	0.20	5,000	7,200	*n/a*	
Averaged RMRP histogram						
$n = 10,000$	0.06	0.22	2	13	811	902
$n = 50,000$	0.03	0.15	15	2,168	1,546	1,719
3-d						
KDE ($n_K = 2,000$)	0.13	0.35	5,600	7,200	*n/a*	
Averaged RMRP histogram						
$n = 10,000$	0.24	0.41	21	451	1,573	1,718
$n = 50,000$	0.12	0.30	295	27,832	3,507	3,783
4-d						
KDE ($n_K = 2,000$)	0.25	0.51	7,200	8,050	*n/a*	
Averaged RMRP histogram						
$n = 50,000$	0.32	0.47	2,524	53,190	6,241	6,570
$n = 100,000$	0.25	0.42	10,382	82,684	9,431	9,775
5-d						
KDE ($n_K = 2,000$)	0.41	0.66	7,350	8,880	*n/a*	
Averaged RMRP histogram						
$n = 50,000$	0.65	0.67	28,841	277,071	9,342	9,803
$n = 100,000$	0.53	0.60	24,244	399,016	15,160	15,563

The SRP MCMC method is still incomplete, notably in the use of the natural Catalan prior instead of a prior that can be properly calibrated to the sample size, but Table 7.2 suggests that, even with these outstanding issues to be resolved, the averaged histogram RMRP created using a reasonably large sample size can provide a closer estimate of a multivariate density than a KDE with a diagonal bandwidth matrix. This is particularly true with respect to the estimate close to the modes of the true density. The key point is that the partition of the SRP histograms sampled from the Markov chain are data-driven and can thus achieve at least

some locally-adaptive smoothing.

Figure 7.5 illustrates this using the same technique of taking 1-dimensional slices through the RMRP estimate and comparing these to 1-dimensional profiles of the true density and the KDE as is used in Figure 7.4 (in Figure 7.4 the RMRP estimate is the approximation to the KDE; in Figure 7.5 it is the averaged SRP histogram RMRP). The ability of the averaged histogram to get closer to the modes of the true density is clearly apparent. A different specification of `GetValue` might improve the approximation of the KDE but the best approximation method cannot correct inaccuracies in the KDE itself. The disadvantages of the RMRP averaged histogram estimate are also illustrated: it is not differentiable and it is much more influenced by the features of individual data sample (the classic bias-variance trade-off (Scott 1992, chap. 3)).



(a) $d = 2$, $x_2 = -1.5$, $n = 50,000$.

(b) $d = 2$, $x_1 = 2.0$, $n = 50,000$.

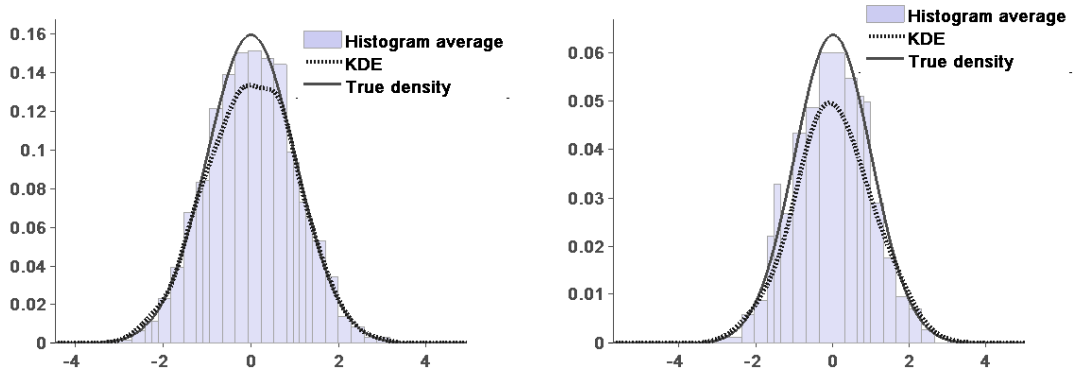
(c) $d = 3$, $x_2 = x_3 = -1.5$, $n = 50,000$.

(d) $d = 3$, $x_1 = x_3 = 2.0$, $n = 50,000$.

Figure 7.5: Density II, KDE and averaged RMRP histogram slice.

A similar comparison of a diagonal bandwidth matrix KDE, an RMRP approximation to the KDE, and averaged RMRP histogram estimate for multivariate data is given in Figure 7.6. The true density here is f_{III}, the standard multivariate Normal density, Example Density III in Appendix B. The 1-dimensional slices of the RMRP averaged histogram density estimates

and the profiles of the true density and the KDE are taken through the mode μ. Again the oversmoothing in the KDE at around the mode is clearly apparent and again the averaged RMRP histogram is less susceptible to this but is also undersmoothed away from the mode.



(a) $d = 2$, $x_2 = 0$, $n = 50,000$.

(b) $d = 3$, $x_2 = x_3 = 0$, $n = 50,000$.

Figure 7.6: Standard multivariate Normal density, KDE and averaged RMRP histogram slice.

7.7 Conclusion

The results in this chapter suggest that the potential for approximating KDEs using RMRPs in order to combine some of the faster convergence of the KDE with efficient operations on the RMRP structure should be explored further.

A multivariate KDE method will provide some sort of estimate even using an inadequate data set and unsuitable bandwidths. The approximation obtained using `RPQApproximate` (Algorithm 7.1) will be affected by any flaws in the KDE being approximated, whether these are due to the bandwidth matrix chosen, the choice of kernel, or the sample data used. The choice of bandwidth matrix in particular has been shown to be crucial to the performance of a multivariate KDE (Scott and Sain 2005). Many KDE methods are unsuitable for large data sets, and as the number of dimensions increases not only do the data sets required get larger but the use of a fixed bandwidth or diagonal bandwidth matrix may result in severe oversmoothing around the modes of the target density.

If the KDE is misleadingly inaccurate, the RMRP approximation to it will be similarly inaccurate. When the true underlying distribution is completely unknown and the inadequacies of the KDE are not obvious there may be some risk that the ease with which the approximation can be formed could mask the need to ensure that the KDE method itself, and the sample size, are appropriate for the data dimensions.

The investigations described in this chapter used only a single kernel density estimator, the MCMC bandwidth estimator of Zhang et al. (2006). This uses a diagonal bandwidth

matrix, i.e., one bandwidth for each coordinate of the data. The MCMC bandwidth selection process means that this estimator is very computationally intensive, although it has also been shown to cope relatively well, compared with other popular bandwidth selection methods, with multivariate data in up to five dimensions (Zhang et al. 2006).

The results discussed in this chapter suggest that, in two to three dimensions, an SRP-based averaged histogram RMRP estimate created using the SRP MCMC method of Chapter 6 may have advantages over an RMRP approximation to a KDE created using this particular kernel density estimator, *provided that* sufficient sample data is available to be able form a reasonable SRP MCMC estimate. This is a crucial caveat. The SRP MCMC method requires at least an order of magnitude more data than the KDE (for example, at least about 10,000 data points for two-dimensional data and at least about 50,000 data points for three dimensional data, compared with 1,000–2,000 data points for the KDE).

The investigations described in this chapter show that, when sufficient data is available to form the SRP MCMC estimate, this estimate benefits from the locally-adapted smoothing that results from data-driven SRP partitioning, while the diagonal bandwidth KDE of Zhang et al. (2006) is more vulnerable to the problem of oversmoothing around the modes.

The next step should therefore be to compare the SRP MCMC method to RMRP approximations of a KDE using some form of locally-adaptive bandwidth matrix. The main issue with such estimators seems to be that, in practice, most are only suitable for univariate or bivariate data (Sain 2002; Gray and Moore 2003a), but if such an estimator could be found with reasonable running times for three or four-dimensional data, then it seems likely that an RMRP KDE approximation will then be a much better density estimate than the RMRP formed from the SRP MCMC process. As is noted in this chapter and in Chapter 6, the SRP MCMC process as currently implemented is already struggling with the size of the data sample required to form an adequate density estimate in $d = 4$.

Another interesting extension of this research would be to compare marginal and conditional density estimates computed from an RMRP created using the SRP MCMC method with those created using the quick multivariate kernel density estimator of Cheng et al. (2006). This kernel density estimator is suitable for large data sets, and so exactly the same sample data could be used for each estimate, and exploits the intermediate results of its data binning algorithm to provide a means to create marginal and conditional density estimates quickly, without repeating the entire density estimation process.

In higher dimensions (about $d = 5$ and above) the SRP MCMC method as currently implemented requires impractically long running times to achieve convergence. A KDE approximation could provide the only feasible means of obtaining a density estimate represented as an RMRP in these dimensions. However, as is noted in Section 2.5, only a very limited number of KDE methods are able to give reasonable estimates in this number of dimensions and above.

Further research could also investigate the possibility of using an RMRP approximation to

a KDE as a way of calibrating the temperature of the prior used for the MCMC process. An RMRP approximation to a KDE might also provide another way to get an accelerated starting point for the MCMC process.

The most interesting advance of the preliminary investigation in this chapter will probably be to try to use fast high-dimensional KDE methods, such as those discussed in Lee et al. (2006) and (Raykar et al. 2010), with the `RPQApproximate` operation (Algorithm 7.1). Future work on KDE approximation should also investigate different ways to use the KDE to map a value to each node in the RMRP (`GetValue`) and alternative priority functions. The question of how to determine appropriate values for $\overline{\psi}$ and $\overline{m}$ will also need to be addressed.

Chapter 8

Simulation-intensive inference

8.1 Complex models and intractable likelihoods

Section 2.1 described classical and Bayesian inference using the likelihood function or Bayesian posterior. Parametric inference for θ using these approaches seems to depend on being able to specify an expression for the likelihood function $\mathcal{L}(\theta)$, and being able to evaluate it. In some cases the model is restricted to some family of distributions $\mathfrak{F}$ chosen so that a *tractable* likelihood (Hartig et al. 2011) is available.

In many situations, however, the models required are highly complex and it is hard, or impossible, to calculate the likelihood directly. In fields such as ecology, biology, and genetics stochastic models are often built up using a large number of interacting processes or sub-models (Beaumont 2010; Bertorelle et al. 2010; Hartig et al. 2011). These models are impossible to analyse in their entirety but are often relatively easy to simulate from using a computer (Marjoram and Tavaré 2006). Given an implementation of the model dependent on theta, $f(x, \theta)$, and a particular parameter value $\theta^\star$, simulated data x_{sim} can be generated from $f(x \mid \theta^\star)$. The remainder of this chapter discusses how simulations can be used to support inference for complex models with intractable likelihoods.

An important point about modelling and inference in these complex situations is that the data itself is typically very high dimensional. Usually summary statistics of the data are used to reduce the dimensions of the problem. For example, in population genetics the 'data' may be the nucleotide types at thousands (or more) sites on the chromosomes of a sample of present-day individuals. This is summarised with low-dimensional statistics (Beaumont 2010). Ideally the statistics used would be sufficient summaries of the data in the context of the inferences to be made; in practice it is usually accepted that the summary statistics employed are not sufficient, and it may be difficult even to identify which of the available statistics is the most useful (Bertorelle et al. 2010). Thus there is an approximation inherent in trying to find the likelihood or posterior given summary statistics of the observed data rather than the observed data itself (Marjoram and Tavaré 2006). In this thesis t is used to represent some scalar or vector summary of the data. An observation or realisation $x^\star$ is summarised as $t^\star$.

The simplest use of simulation is 'what if', or investigating the effects of different model details, assumptions, or parameter values. Given some $\theta^\star$, simulations x_{sim} from $f(x; \theta^\star)$ can be summarised or used to create a density estimate for $f(t; \theta^\star)$. When the model interactions are complex, and t and θ have a large number of dimensions, gaining useful insights even from this simple approach may be difficult (Hartig et al. 2011). Low dimensional summaries of the results may hide important features while at the same time it becomes harder to create reliable

high-dimensional estimates of the whole density. A large number of combinations of effects may also have to be investigated to reveal the effects of interactions between the different parameters.

Using simulations for parametric inference (model comparison or parameter estimation) is more problematic: simulating large amounts of data from the model does not seem to solve the problem of not being able to specify a likelihood function for θ.

A simple approach using a form of simulation as the basis for inference is nonparametric hypothesis testing. Permutation tests use random permutations of the observed data to build a simulated sample and the P-value is calculated as the proportion of simulated values at least as extreme as the summary statistic of the observed data (Hudson et al. 1992; Excoffier and Lischer 2010). A similar approach simulates from the results of a Fisher Exact Test under the null distribution using a Markov chain of contingency tables (Raymond and Rousset 1995).

The permutation and exact tests can be carried out without implementing a model of the underlying process (the 'simulations' come from manipulations of the observed data), but can only be used for a limited range of inference problems. The amount of the observed data may also limit the ability of the tests to be able to identify a significant result: for small amounts of data the number of possible permutations is limited (Fitzpatrick 2009).

More sophisticated approaches to simulation-intensive inference are discussed in the remaining sections of this chapter. Many of these techniques use a Bayesian approach. This fits naturally with the simulation of data. Values for the model parameters θ used in the simulation process are drawn from a prior distribution $f(\theta)$ and used to simulate data from the sampling density $f(x \mid \theta)$. The target is the posterior density $f(\theta \mid t^\star)$ of θ given the summary statistic $t^\star$ of the observed data $x^\star$.

8.2 Approximating the likelihood function

A wide range of simulation-based methods may be used to assess the 'goodness-of-fit' of some specific, parameterised, model. In simple cases a brute force approach can be used, simulating sufficient data from the model to calculate an estimate of the density at $t^\star$ as the relative proportion of the simulations equal to $t^\star$. If the summary statistics are continuous then the likelihood may approximated using the relative frequency of simulations with summary statistic close to $t^\star$ (Hartig et al. 2011) or by using the simulations to create a density estimate and evaluating that estimate at $t^\star$. The simulated maximum likelihood method of Tian et al. (2007) includes the latter approach. The authors suggested that a kernel density estimator could be used to form the required density estimate for one-dimensional data, or "the theory of multivariate density estimation (Scott, 1992)" (Tian et al. 2007, p. 85) for multivariate data; the example actually given in the paper used a discrete-valued molecular number for the summary statistic t and estimated the probability mass function evaluated at an observed $t^\star$

as the relative frequency of $t^\star$ in the simulations.

As usual, dimensionality matters: the sample size needed grows with the number of dimensions of the summary statistic. In some approaches assumptions are made about the distribution of the summary statistics. Simulations from the model can then be used only to make estimates of the parameters for the distribution rather than to estimate the density itself, and classical likelihood ratio tests can be used to compare models using parametric likelihood approximations (for example, Wood (2010)). Other simulation-based goodness of fit measures or model selection methods described in Hartig et al. (2011) avoid the problem of approximating the likelihood by essentially substituting a single goodness-of-fit test not based on the likelihood (such as the distance between the mean simulated summary statistic and $t^\star$) or by using a combination of rejection filters (each usually based on a relatively simple comparison of some statistic of the simulated sample to some required value or range of acceptable values based on $t^\star$).

When the parameter space is sufficiently small, an approximate likelihood function itself may be constructed by interpolating between pointwise estimates of $\mathcal{L}(\theta^\star)$ for particular values of $\theta^\star$. Another simulation-based approach is to try to sample directly from the likelihood function $\mathcal{L}(\theta)$ or from the Bayesian posterior $f(\theta \mid x^\star)$ or $f(\theta \mid t^\star)$. Rejection sampling methods sample random 'proposed' values for θ from the prior and accept or reject each proposal θ' depending on the value of a pointwise approximation of $\mathcal{L}(\theta')$. This method can be very computationally expensive, possibly requiring a density estimate to be created from simulated t for each proposal θ' in order to approximate $\mathcal{L}(\theta')$.

The term 'synthetic likelihood', attributed to Wood (2010), is sometimes used to describe methods to simulate or synthesis the likelihood function. The example described in Wood (2010) relies heavily on the use of a likelihood-friendly multivariate Normal model. The vector of summary statistics chosen is modelled as a multivariate Normal random variable. For a given θ, the parameters of this density are estimated from the mean and covariance matrix of the summary statistics for a sample of data simulated using that θ. The synthetic likelihood of θ is then defined to be the likelihood of a multivariate Normal model with the estimated parameter values using the usual expression for a multivariate Normal likelihood. This process gives a pointwise synthetic likelihood for a given θ; a sample of values for θ is obtained by using the synthetic likelihood in the acceptance step of a Monte Carlo Markov chain.

MCMC algorithms attempt to speed up the time that it takes to simulate enough samples of θ from $\mathcal{L}(\theta)$ or $f(\theta \mid x^\star)$ to be able to make an estimate of the shape of the likelihood function or posterior by sampling most from the areas of higher likelihood (Hartig et al. 2011). Assessing whether the chain has converged (is sampling from the target distribution) is not straightforward (Marjoram and Tavaré 2006). MCMC methods are also prone to becoming trapped in areas of low likelihood (Bertorelle et al. 2010) and to false convergence around local

maxima, although methods to force the chain to explore the parameter space can be added to mitigate this (Marjoram and Tavaré 2006).

Importance sampling can also be used to try to make the sampling process more efficient. Kuhner et al. (2000) describe a multi-stage approach for likelihood-based inference for population genetics using a Markov chain to create a sample of high-likelihood genealogies based on initial estimates of the parameter values of interest, which are then used to create a sample of parameter values to which a likelihood surface is fitted and a maximum likelihood estimate is made; several iterations are usually used and various technical difficulties may be encountered in finding a maximum likelihood estimate (Kuhner and Smith 2007). A Bayesian version of this process is also available (Kuhner 2006).

Sequential Monte Carlo methods are another attempt to speed up sampling. These are particle filters, sampling multiple θ' in each step and assigning each a weight proportional to its (pointwise approximate) likelihood or posterior value (Hartig et al. 2011).

There are a number of technical issues involved in avoiding bias when using these efficient sampling methods to estimate the likelihood function or posterior distribution ((Hartig et al. 2011; Beaumont et al. 2009)).

8.3 Approximate Bayesian computation

Section 8.2 describes various Bayesian methods, including rejection sampling, MCMC, and sequential MCMC, that use pointwise approximations to the likelihood for a particular $\theta^\star$ to construct an estimate the posterior $f(\theta \mid t^\star)$. A nonparametric pointwise approximation of the likelihood can be made using a relative frequency distribution or density estimate created from a sample of summary statistic values simulated under each proposal, as described above, but this is typically very computationally expensive.

Approximate Bayesian computation (ABC) is the term used for a group of Bayesian methods that seek to avoid this source of computational complexity entirely. Under the ABC approach the need for an explicit pointwise estimate of the likelihood function is bypassed entirely by implicitly building it into the decision rule for accepting new parameter values. ABC is one of the most active areas of research into efficient sampling for simulation-intensive inference (Beaumont 2010; Bertorelle et al. 2010).

Ideally, a proposed θ' is used to simulate data summarised as t' and is accepted if $t' = t^\star$, so that the acceptance probability is proportional to the likelihood ($P(\text{accepting } \theta') \propto \mathcal{L}(\theta')$) and the sample built up approximates the target $f(\theta \mid t^\star)$. Because the summary statistics are usually continuous random variables the acceptance step usually requires $|t' - t^\star| \leq \varepsilon$ rather than $t' = t^\star$, $|.|$ being some appropriate measure of the distance between t' and $t^\star$. There are ABC versions of the algorithms for rejection sampling, MCMC sampling, and sequential Monte Carlo sampling.

ABC rejection sampling can still be too inefficient for many applications and convergence using ABC-MCMC methods may also be slowed by poor mixing in the tails of the distribution (Beaumont 2010). In any algorithm using ABC there is a trade-off between the approximation error that results from the use of $|t' - t^\star| \leq \varepsilon$ rather than $t' = t^\star$ and sampling speed: a larger ε speeds up sampling but increases the bias of the resulting density estimate towards the prior. Where the summary statistics are multi-dimensional a larger ε will be needed to be able to achieve reasonable sampling times.

To try to deal with these issues it is common to keep a larger tolerance ε in the ABC procedure and add a final conditional density estimation step to try to improve the approximation. This may be achieved, for example, by using local regression to adjust the sampled parameter values to reduce the influence of the prior distribution (Beaumont et al. 2002). The regression step itself is subject to problems with high-dimensional summary statistics and computational inefficiency (Blum and François 2010) and enhanced versions of this step have been proposed to try to find a more efficient way to calculate a final sample of adjusted parameter values and mitigate the problems associated with ε, bias, and the approximation error (Beaumont 2010; Bertorelle et al. 2010). These methods have advantages compared with the original local regression proposed by Beaumont et al. (2002) but are also considerably more complicated to implement (for example, the neural network algorithm used by Blum and François (2010)) and/or involve more parametric assumptions (for example, the generalised linear modelling approach of Leuenberger and Wegmann (2010)).

The summary statistics used in the algorithms described above have also received a large amount of attention from ABC researchers. Until recently the summary statistic was typically seen as a low-dimensional compression of the data, the ideal being a sufficient statistic (lossless compression, with regard to information relevant to the parameters of interest). Transformations of the summary statistics (for example, using principle components analysis or partial least-squares) may be used to reduce correlations between the components and reduce dimensionality (Ray et al. 2010). Fearnhead and Prangle (2012) suggest a rather different approach where the summary statistic is some function of the data that minimises some error loss in the process overall. For L_2 error loss the appropriate summary statistic is the expectation of the posterior distribution of θ. This summary statistic must be estimated using an initial simulation (for example, an initial ABC process with some almost-arbitrary choice of summary statistics) followed by a regression step to model this as a function of user-selected transformations of the data. The ABC process is then re-run using the estimated summary statistic.

ABC methods are used for model choice (most straightforwardly with rejection or sequential Monte Carlo sampling) as well as for inferences about parameter values. For model choice, an index for the model may be treated as a categorical variable corresponding to an element of θ, leading to a calculation of a 'posterior probability' for each model based either on the

number of acceptances from each model or a regression of simulated summary statistics on the model index as the regression response variable (Beaumont 2008; Cornuet et al. 2008; Guillemaud et al. 2009). However, summary statistics that are sufficient for the parameter estimation questions (*modelwise sufficient*) may not be sufficient for the model choice question (*sufficient across models*), and when the summary statistics used are not sufficient the sampling process may not converge to the target distribution (Robert et al. 2011).

Overall, recent review papers such as Bertorelle et al. (2010), Csilléry et al. (2010), and Hartig et al. (2011) see many advantages in ABC but also point out the complexities and possible weaknesses of this family of methods. Partially in response to the growing complexity of the algorithms, several impressive software packages have been developed to make it as user-friendly as possible (for example, Cornuet et al. (2008) and Wegmann et al. (2010)). However, both Bertorelle et al. (2010) and Csilléry et al. (2010) allude to the potential problems that may result from hiding the complexities and uncertainties of the process, and the importance of some of the underlying assumptions, from the user.

8.4 Adaptive approximate Bayesian computation

Adaptive approximate Bayesian computation (AABC) tries to deal with the limitations of rejection-ABC and ABC-MCMC methods by increasing the opportunities within the process to 'learn' and use that learned information (adapt) to get a faster and more accurate estimate of the target posterior distribution (Beaumont 2010; Bertorelle et al. 2010). Introducing adaptation into an MCMC approach without disrupting the convergence characteristics of the chain is not straightforward, but can be used with an importance sampling approach. Adaptation is achieved by sampling iteratively or sequentially.

The versions of AABC proposed by Sisson et al. (2007) and Beaumont et al. (2009) incorporate a rejection sampling element. Each value θ' for θ proposed for inclusion in the sample is used to simulate data x' and hence a summary statistic t'. Only values of θ' associated with t' such that $|t' - t^\star| \leq \varepsilon$ are accepted. In theory, by using a decreasing series of tolerances $\{\varepsilon_j\}$ with ε_j being the tolerance for iteration $j = 0, 1, \ldots$, the methods may achieve both reasonable acceptance rates during each iteration and a final sample that gives a close enough approximation of the target posterior density. In practice, however, it may be found that as ε_t decreases the acceptance ratio becomes very very small without any major improvement in the approximation to the posterior. To avoid this the AABC process can be carried out using relatively generous values of ε_j, followed by some form of conditional density estimation, for example a local regression step or similar adjustment of the samples (Beaumont 2010).

This short summary of AABC omits many of the technical aspects and theoretical considerations (Cappé et al. 2004; Douc et al. 2007; Sisson et al. 2007; Beaumont et al. 2009) that need to be taken into account when developing AABC methods. There is considerable potential for

a flawed design to result in biased or otherwise unreliable results. However, interest in AABC is spreading beyond the fields of ecology, biology and genetics where it is most commonly employed. For example, Cameron and Pettitt (2012) describe the development and testing of an AABC approach for astronomical model analysis. This includes a comparison of results using summary statistics calculated by an adaptation of the semi-automatic method of Fearnhead and Prangle (2012) with those using more conventional summaries of the data. Cameron and Pettitt (2012) provides a good example both of the potential for the use of AABC methods and of the complexity of the process.

8.4.1 Summary of an adaptive approximate Bayesian computation process

This section gives a summary of the AABC method of Sisson et al. (2007) and Beaumont et al. (2009) using the notation developed in this thesis.

The observed data $x^\star$ is summarised, using some appropriate summary statistics, as $t^\star$. The parameters of interest are denoted by θ. The aim of the process is to form an estimate of the posterior $f(\theta \mid t^\star)$. A model $f(x \mid \theta)$ for the data conditional on θ is constructed. This model together with the method of calculating the summary statistic effectively gives a model $f(t \mid \theta)$ for summary statistics conditional on θ. A prior $f(\theta)$ for θ is specified. A sequence of tolerance values $\{\varepsilon_j\}_{j\in\{1,\ldots,J\}}$ is selected, where J is the total number of iterations of the sampling/resampling process to be used. The number of simulation samples n required to estimate the target posterior, $f(\theta \mid t^\star)$, is specified.

In the first iteration ($j = 1$), values of θ are drawn from the prior $f(\theta)$, $\theta' \sim f(\theta)$. Each θ' drawn is used to generate $x' \sim f(x \mid \theta')$ and x' is summarised as t'. The proposal θ' is accepted as a sample from the approximated posterior if $|t' - t^\star| \leq \varepsilon_1$. This process of proposing and evaluating $\theta' \sim \boldsymbol{f}(\theta)$ continues until n values have been accepted. These n values of θ are regarded as being drawn from an approximation $f_{\varepsilon_1}(\theta \mid t^\star)$ of the target posterior $f(\theta \mid t^\star)$. In subsequent iterations $j = 2, \ldots, J$ the values of θ are drawn from $f_{\varepsilon_{j-1}}(\theta \mid t^\star)$ using an importance sampling weight to adjust for the fact that the samples are not drawn from the prior $f(\theta)$. In the detailed process described by Beaumont et al. (2009) a forward transition kernel is used to move the value θ' drawn by a small amount away from the value actually resampled. This is a refinement to the process and is not described in more detail in this summary. Again each θ' drawn is used to generate $x' \sim f(x \mid \theta')$, x' is summarised as t', and θ' is accepted as a sample from the approximated posterior if $|t' - t^\star| \leq \varepsilon_j$. Sampling and evaluation continues until n values for θ have been collected. The final result is a sample drawn from an approximation $f_{\varepsilon_J}(\theta \mid t^\star)$ to the target posterior $f(\theta \mid t^\star)$. Local regression or some other form of conditioning may then be used to try to improve the closeness of the approximation.

8.5 The observed data and approximate Bayesian computation

ABC methods are therefore typically approximate in two senses: the use of possibly insufficient summary statistics of the data, and the use of the tolerance ε. The theoretical target is $f(\theta \mid x^\star)$, the target in practice is $f(\theta \mid t^\star)$ (see Section 8.1). The actual estimate made is an approximation to $f(\theta \mid t^\star)$ because $P(|t' - t^\star| \leq \varepsilon)$ is used to approximate $f(t^\star \mid \theta') \propto \mathcal{L}(\theta')$ when determining whether to accept a draw θ' as a sample from $f(\theta \mid t^\star)$ on the basis of the summary statistic t' of data simulated using θ' (see Section 8.3).

An important point to note is that ABC methods treat the summary statistics of the observed data as a single unit of summary statistics $t^\star$. It is relatively easy to build the approximation $P(|t' - t^\star| \leq \varepsilon)$ for a single $t^\star$ into a process for sampling from a ε-dependent approximation to $f(\theta \mid t^\star)$. It is less straightforward to replace a product likelihood

$$\mathcal{L}(\theta \mid t_1^\star, \ldots, t_{n_{obs}}^\star) \propto f\left(t_1^\star, \ldots, t_{n_{obs}}^\star \mid \theta\right) = \prod_{i=1}^{n_{obs}} f(t_i^\star \mid \theta) \quad ,$$

where n_{obs} is the number of independent samples of the data available from which to calculate the summary statistics. For example, in applications of AABC to population genetics, when information for several loci is available the summary statistics are often the means and variances of the statistics for the individual loci (Bertorelle et al. 2010).

It might be possible to use independent samples to obtain n_{obs} approximate posteriors $f(\theta \mid t_i^\star), \ldots, f\left(\theta \mid t_{n_{obs}}^\star\right)$ and make an informal assessment of the possible form of $f\left(\theta \mid t_i^\star, \ldots, t_{n_{obs}}^\star\right)$ from these. More rigorously, sequential runs of the ABC algorithm could be used to emulate a Bayesian updating scheme. The estimate of $f(\theta \mid t_1^\star)$ from the first run would be used as the prior density to be used in conjunction with $t_2^\star$ to obtain an estimate of $f(\theta \mid t_1^\star, t_2^\star)$, which then be used as the prior for the next run, etc. There are a number of difficulties with this, the most serious of which would probably be the accumulated effect of the approximations in the posterior estimate resulting from using $|t - t_i^\star| \leq \varepsilon$ instead of $t = t_i^\star$ in each run. The time taken for the entire process would also increase.

The AABC literature reviewed for this thesis consistently treated the observed data as a single unit. In some applications it may not be possible to obtain multiple independent data samples, or it may be possible to find a statistic that provides a sufficient summary of the independent sample. In other cases independent samples may be available but a single unit of summary statistics is used because this is a requirement of the ABC algorithm. In population genetics (one of the most active fields for ABC and AABC methods), for example, the genetic ancestries of different loci on the same chromosome of the same individual may be regarded as independent of each other, due to recombination, so that samples of genetic data taken from these different loci can be regarded as independent samples (Wakeley 2009, chap. 7).

Chapter 9

RPABC: Regular pavings for simulation-intensive inference

9.1 Introduction

This chapter describes how RP-structured density estimates can be used for simulation intensive inference. The method is referred to as RPABC. RPABC exploits the operational properties of RP-structured density estimates and the relationship between posterior densities, joint densities, and conditional densities. These relationships are discussed in Chapter 2 and are briefly reviewed below in Section 9.2. Section 9.3 gives an overview of RPABC.

Sections 9.4, 9.5, 9.6 and 9.7 contain examples of RPABC applied in two different contexts. In each case an AABC approach and the results of a hybrid method, combining both AABC and RPABC, are also described. The AABC results are illustrated with RMRP density estimates created using the SRP MCMC method described in Chapter 6. This choice of visualisation is motivated partly to facilitate comparison with the RPABC results but also because the operational capabilities of RMRPs are useful for visualising and manipulating the AABC posterior density estimates.

9.2 The posterior density reviewed

Classical analytical Bayesian methods formulate the posterior density for a parameter θ given some observed data $x^\star$ summarised by $t^\star$ as $f(\theta \mid t^\star) = \frac{f(t^\star \mid \theta) f(\theta)}{f(t^\star)}$. ABC methods bypass the need for an analytical expression for $f(t^\star \mid \theta)$ by attempting to simulate directly from $f(\theta \mid t^\star)$ and using the simulations to construct a density estimate $\hat{f}(\theta \mid t^\star)$.

An alternative approach to constructing an estimate $\hat{f}(\theta \mid t^\star)$ is to first estimate the joint density $f(\theta, t)$ and obtain an estimate of the conditional density $f(\theta \mid t^\star)$ directly from the joint estimate. Assuming that the joint density is estimated as the product of a prior density $f(\theta)$ for θ and a model $f(t \mid \theta)$ for the summary statistics conditional on the value of θ, $f(\theta, t) = f(t \mid \theta) f(\theta)$, then the estimate of the conditional density $f(\theta \mid t^\star)$ is an estimate $\hat{f}(\theta \mid t^\star)$ of a Bayesian posterior density.

Conditional density estimates and marginal densities can also be combined to form an estimate of a posterior $f\left(\theta \mid t_1^\star, \ldots, t_{n_{obs}}^\star\right)$. This is most easily explained in relation to classical Bayesian updating. Given one summary statistic $t_1^\star$ the posterior density is $f(\theta \mid t_1^\star)$. Given $t_2^\star$ independent of $t_1^\star$,

$$f(\theta \mid t_1^\star, t_2^\star) \propto f(t_2^\star \mid \theta) f(\theta \mid t_1^\star) \quad , \tag{9.1}$$

and given $t_3^\star$ independent of $t_1^\star, t_2^\star$,

$$f\left(\theta \mid t_1^\star, t_2^\star, t_3^\star\right) \propto f\left(t_3^\star \mid \theta\right) f\left(\theta \mid t_1^\star, t_2^\star\right) \ ,$$

and the sequential updating process can be carried on for $t_3^\star, \ldots, t_{n_{obs}}^\star$ to obtain

$$f\left(\theta \mid t_1^\star, \ldots, t_{n_{obs}}^\star\right) \propto f\left(t_{n_{obs}}^\star \mid \theta\right) \cdot f\left(t_{n_{obs}-1}^\star \mid \theta\right) \cdots f\left(t_2^\star \mid \theta\right) \cdot f\left(\theta \mid t_1^\star\right) \ .$$

Alternatively, first estimate the joint density $f(\theta, t)$ and then, given independent values $t_1^\star$ and $t_2^\star$, construct density estimates of the two conditional (posterior) densities $f\left(\theta \mid t_1^\star\right)$ and $f\left(\theta \mid t_2^\star\right)$. Then

$$\begin{aligned} f\left(\theta \mid t_2^\star\right) f\left(\theta \mid t_1^\star\right) &= \frac{f\left(t_2^\star \mid \theta\right) f(\theta)}{f(t_2^\star)} f\left(\theta \mid t_1^\star\right) \\ &\propto f\left(t_2^\star \mid \theta\right) f(\theta) f\left(\theta \mid t_1^\star\right) \ , \end{aligned}$$

and, noting that $f\left(\theta \mid t_1^\star, t_2^\star\right) \propto f\left(t_2^\star \mid \theta\right) f\left(\theta \mid t_1^\star\right)$ (Equation (9.1)),

$$\begin{aligned} f\left(t_2^\star \mid \theta\right) f(\theta) f\left(\theta \mid t_1^\star\right) &= f\left(t_2^\star \mid \theta\right) f\left(\theta \mid t_1^\star\right) f(\theta) \\ &\propto f\left(\theta \mid t_1^\star, t_2^\star\right) f(\theta) \ , \end{aligned}$$

and thus

$$f\left(\theta \mid t_1^\star, t_2^\star\right) \propto \frac{f\left(\theta \mid t_2^\star\right) f\left(\theta \mid t_1^\star\right)}{f(\theta)} \ .$$

This can be extended for n_{obs} independent values $t_1^\star, \ldots t_{n_{obs}}^\star$ to obtain

$$f\left(\theta \mid t_1^\star, \ldots, t_{n_{obs}}^\star\right) \propto \frac{\prod_{i=1}^{n_{obs}} f\left(\theta \mid t_i^\star\right)}{\left(f(\theta)\right)^{n_{obs}-1}} \ . \tag{9.2}$$

The Bayesian posterior predictive density of a new observation given $t_1^\star, \ldots, t_{n_{obs}}^\star$ is a conditional density $f\left(t \mid t_1^\star, \ldots, t_{n_{obs}}^\star\right)$ obtained by marginalising the product of the posterior and the model:

$$f\left(t \mid t_1^\star, \ldots, t_{n_{obs}}^\star\right) \propto \int f\left(t \mid \theta\right) f\left(\theta \mid t_1^\star, \ldots, t_{n_{obs}}^\star\right) \mathrm{d}\theta \ .$$

9.3 Regular paving approximate Bayesian computation

9.3.1 The observations, the model, and the prior

The observed data $x_1^\star, \ldots, x_{n_{obs}}^\star$, assumed to be n_{obs} independent realisations, is summarised using some appropriate summary statistics, as $t_1^\star, \ldots, t_{n_{obs}}^\star$. The parameters of interest are

denoted by θ. The aim of the process is to form an estimate of the posterior $f\left(\theta \,|\, t_1^\star, \ldots, t_{n_{obs}}^\star\right)$. A model $f\left(t \,|\, \theta\right)$ for the summary statistics conditional on the parameter(s) of interest θ is constructed. θ may be a vector. A prior $f(\theta)$ for θ and the number of simulation samples n required to estimate the target posterior are specified.

One of the drawbacks of RPABC, in comparison with AABC, is that it may be problematic to identifying a suitable support for the prior. A large prior support is theoretically desirable (so that interesting information in the tails of the posterior density is not inadvertently lost), but the computational efficiency of the method is affected by the size of the joint (parameter and summary statistic) sample space, and the parameter space is determined by the support of the prior. If the support of the prior includes large sets of values of θ where the density of $f\left(t_i^\star \,|\, \theta\right)$ is very low or equal to zero for all of the summary statistics $t_1^\star, \ldots, t_{n_{obs}}^\star$ of the observed data, then many of the simulations used to estimate the joint density $f\left(\theta,\, t\right)$ are essentially wasted. If simulation from the model $f\left(t \,|\, \theta\right)$ is computationally expensive, as it often is for the complex models that require the use of simulation intensive inference methods, using too wide a prior can increase the computation time considerably without adding useful information to the final outcome.

The efficiency of AABC methods is less sensitive to a wide prior support because parameter values are only drawn from the prior itself in the first iteration. In subsequent iterations parameter values are drawn from those which have already generated at least one summary statistic reasonably close to the summary statistic of the observed data.

To mitigate this problem a heuristic method to make a conservative guess at an appropriate support for the prior using the observed data and a wide preliminary prior support was developed for the implementation of RPABC used to produce the results shown in this chapter. This step in the process is similar to a modified version of a single, preliminary, AABC iteration with a large tolerance ε, using the observed data to guide the range of parameter values from which to sample next. The method is described in detail in Appendix K.

9.3.2 Estimating the joint density

The joint density of the parameters and summary statistics is

$$f\left(\theta,\, t\right) = f\left(t \,|\, \theta\right) f(\theta) \ .$$

A sample of n values is drawn from the joint density $f\left(\theta,\, t\right)$ using a two-step simulation process. A sample of values $\theta^{(1)}, \ldots \theta^{(n)}$ is drawn from the prior $f(\theta)$; each $\theta^{(i)}$, $i = 1, \ldots, n$, is then used to generate a summary statistic $t^{(i)}$ from $f\left(t \,|\, \theta_i\right)$. The result is a set of n tuples $(\theta^{(1)},\, t^{(1)}), \ldots, (\theta^{(n)},\, t^{(n)})$.

The n simulations from the joint density are used to create an RMRP density estimate ${}^{\square}f^{(\theta,t)}$. Any suitable method can be used to create ${}^{\square}f^{(\theta,t)}$. Sections 9.4 and 9.7 shows examples

using an RMRP estimate of the expectation under the posterior distribution of SRP histograms using the Metropolis-Hastings MCMC method discussed in Chapter 6. Sections 9.4 and 9.6 also shows a joint density estimate as an RMRP approximation to a KDE using the method described in Chapter 7.

Let d_p denote the number of parameters of interest and d_s denote the number of summary statistics used to summarise a realisation of the data. The joint simulations $(\theta^{(i)}, t^{(i)})$, $i = 1, \ldots, n$, are $(d_p + d_s)$-dimensional tuples,

$$(\theta^{(i)}, t^{(i)}) = (\theta_1^{(i)}, \ldots, \theta_{d_p}^{(i)}, t_1^{(i)}, \ldots, t_{d_s}^{(i)}) \ .$$

Let $\rho^{(\theta,t)}$ denote the root node of ${}^{\square}f^{(\theta,t)}$ and let $\boldsymbol{x}_\rho^{(\theta,t)} \in \mathbb{R}^{(d_p+d_s)}$ denote the root box of ${}^{\square}f^{(\theta,t)}$. The coordinates of $\boldsymbol{x}_\rho^{(\theta,t)}$ are $\Delta = \{1, \ldots, d_p, d_p + 1, \ldots, d_p + d_s\}$ and

$$\boldsymbol{x}_\rho^{(\theta,t)} = \boldsymbol{x}^{(\theta)} \times \boldsymbol{x}^{(t)}$$

where

$$\boldsymbol{x}^{(\theta)} = \bigboxtimes_{i \in \{1,\ldots,d_p\}} [\underline{\theta}_{\rho,i}, \overline{\theta}_{\rho,i}]$$

is a box representing the parameter space and

$$\boldsymbol{x}^{(t)} = \bigboxtimes_{i \in \{1,\ldots,d_s\}} [\underline{t}_{\rho,i}, \overline{t}_{\rho,i}]$$

is a box representing the support of the summary statistics.

If $(d_p + d_s) = 2$ then ${}^{\square}f^{(\theta,t)}$ can easily be rendered for visual display. If $(d_p + d_s) > 2$ then one and two-dimensional projections of the joint density estimate can easily be created as marginal density estimates using `Marginalise` (Algorithm 3.10, Section 3.6.2.5). For example, a 2-d marginal density function with marginal variables θ_1 and t_1 (the first element of the parameters of interest and the first element of the summary statistic, respectively) can be created using `Marginalise`$(\rho^{(\theta,t)}, \{1, d_p + 1\})$.

Forming the joint density estimate is usually the most computationally intensive part of the process if the SRP MCMC process described in Chapter 6 is used. Although it is usually reasonably fast to do this for very simple two or three-dimensional applications, it may take hours, or days, for higher dimensional problems.

9.3.3 Posterior densities

An RMRP estimate of the posterior density function of θ given the summary statistic $t^\star$ of observed data $x^\star$ can be created using the `MRPSlice` operation (Algorithm 3.7) followed by `Normalise` (Algorithm 3.9) as described in Section 3.6.2.6.

The subset of the coordinates $\Delta = \{1, \ldots, d_p, d_p + 1, \ldots, d_p + d_s\}$ used in the `MRPSlice`

operation is $\Lambda = \{d_p + 1, \ldots, d_p + d_s\}$, the coordinates of the joint tuples that represent the summary statistics. The fixed subtuple point on which to slice is $(t^{\star\dagger}_j)_{j\in\Lambda}$ where $t^{\star\dagger}_j = t^\star_{(j-d_p)}$, the $(j - d_p)^{\text{th}}$ element in $t^\star$.

`MRPSlice`$(\rho^{(\theta,t)}, \{d_p{+}1, \ldots, d_p{+}d_s\}, t^{\star\dagger})$ gives the root node of an RMRP ${}^{\square}f^{|t^{\star\dagger}}$ representing an unnormalised estimate of $f(\theta \mid t^\star)$. Normalising ${}^{\square}f^{|t^{\star\dagger}}$ gives an RMRP representing the posterior density $f(\theta \mid t^\star)$. This RMRP will be denoted by ${}^{\square}f^{\theta|t^\star}$. The root box of ${}^{\square}f^{\theta|t^\star}$ is $\boldsymbol{x}^{(\theta)} = \bigboxtimes_{i\in\{1,\ldots,d_p\}} [\underline{\theta}_{\rho,i}, \overline{\theta}_{\rho,i}]$, the box representing the parameter space.

This process can be used to obtain n_{obs} RMRPs ${}^{\square}f^{\theta|t^\star_1}, \ldots, {}^{\square}f^{\theta|t^\star_{n_{obs}}}$ by slicing using $t^\star_1, \ldots, t^\star_{n_{obs}}$. Each of these RMRPs will have the same root box $\boldsymbol{x}^{(\theta)}$. Equation (9.2) may then be used to obtain an estimate of $f\left(\theta \mid t^\star_1, \ldots, t^\star_{n_{obs}}\right)$ if the prior $f(\theta)$ is also available in the form of an RMRP with root box $\boldsymbol{x}^{(\theta)}$. This can be achieved by again applying `Marginalise` (Algorithm 3.10) to the joint density estimate ${}^{\square}f^{(\theta,t)}$. `Marginalise`$(\rho^{(\theta,t)}, \{1, \ldots, d_p\})$ will give the root node of an RMRP ${}^{\square}f^{\theta}$ with root box $\boldsymbol{x}^{(\theta)}$ as described above. Alternatively, one of the RMRP function approximation algorithms described in Harlow et al. (2012) could be used to obtain an RMRP approximation of the prior density $f(\theta)$ with root box $\boldsymbol{x}^{(\theta)}$.

Provided that $f_{\rho\mathsf{v}} > 0$ for all $\rho\mathsf{v} \in \mathbb{V}\left({}^{\square}f^{\theta}\right)$, arithmetic on RMRPs (Equation (3.10a)) and `Normalise` can then be used to obtain an RMRP ${}^{\square}f^{\theta|t^\star_1,\ldots,t^\star_{n_{obs}}}$ representing an estimate of the posterior density $f\left(\theta \mid t^\star_1, \ldots, t^\star_{n_{obs}}\right)$:

$$ {}^{\square}f^{\theta|t^\star_1,\ldots,t^\star_{n_{obs}}} = \texttt{Normalise}\left(\frac{\prod_{i=1}^{n_{obs}} {}^{\square}f^{\theta|t^\star_i}}{\left({}^{\square}f^{\theta}\right)^{(n_{obs}-1)}}\right) . \tag{9.3} $$

Once the joint density estimate ${}^{\square}f^{(\theta,t)}$ has been formed, the time taken to create each conditional density estimate is minimal, and the multiplication and division operations are also very efficient because of the RP tree structure.

Samples drawn from ${}^{\square}f^{\theta|t^\star_1,\ldots,t^\star_{n_{obs}}}$ using `SimulateData` (Algorithm 3.12, Section 3.6.2.11) can be used as the basis for further inference (Bolstad 2010, chap. 3) about θ.

9.3.4 Highest posterior density regions

The `CoverageRegion` operation (Algorithm 3.11, Section 3.6.2.10) can be used to obtain highest posterior density regions from ${}^{\square}f^{\theta|t^\star_1,\ldots,t^\star_{n_{obs}}}$ or from marginalisations of ${}^{\square}f^{\theta|t^\star_1,\ldots,t^\star_{n_{obs}}}$.

9.3.5 The posterior predictive density

Simulation from ${}^{\square}f^{\theta|t^\star_1,\ldots,t^\star_{n_{obs}}}$ using `SimulateData` (Algorithm 3.12) and `Marginalise` (Algorithm 3.10) can also be used to create an RMRP estimate of the posterior predictive density $f\left(t \mid t^\star_1, \ldots, t^\star_{n_{obs}}\right)$.

As with the creation of the posterior estimate itself, the process starts by creating an RMRP estimate of the joint density $f\left(t,\, \theta \,|\, t_1^\star, \ldots, t_{n_{obs}}^\star\right)$ using simulations. If the size of the simulation sample required for this estimate is n', a sample $\theta^{(1)}, \ldots \theta^{(n')}$ is simulated using n' independent applications of $\texttt{SimulateData}\left({}^{\square}f^{\theta|t_1^\star,\ldots,t_{n_{obs}}^\star}\right)$. Each $\theta^{(i)}$, $i = 1, \ldots, n'$, is then used to generate a summary statistic $t^{(i)}$ from $f\left(t \,|\, \theta^{(i)}\right)$. The result is a set of n' tuples $(\theta^{(1)},\, t^{(1)}), \ldots, (\theta^{(n')},\, t^{(n')})$.

The n' simulations are used to create an RMRP density estimate ${}^{\square}f^{(\theta,t)|t_1^\star,\ldots,t_{n_{obs}}^\star}$. Again, any suitable method can be used including SRP histogram averaging or an RMRP approximation to a KDE. Let $\rho^{(\theta,t)|t_1^\star,\ldots,t_{n_{obs}}^\star}$ denote the root node of ${}^{\square}f^{(\theta,t)|t_1^\star,\ldots,t_{n_{obs}}^\star}$ and let $\boldsymbol{x}_\rho^{(\theta,t)|t_1^\star,\ldots,t_{n_{obs}}^\star} \in \mathbb{R}^{(d_p+d_s)}$ denote the root box of ${}^{\square}f^{(\theta,t)|t_1^\star,\ldots,t_{n_{obs}}^\star}$.

The coordinates of $\boldsymbol{x}_\rho^{(\theta,t)|t_1^\star,\ldots,t_{n_{obs}}^\star}$ are $\Delta = \{1, \ldots, d_p, d_p+1, \ldots, d_p+d_s\}$. An RMRP estimate ${}^{\square}f^{t|t_1^\star,\ldots,t_{n_{obs}}^\star}$ of the posterior predictive density is obtained using `Marginalise`:

$$ {}^{\square}f^{t|t_1^\star,\ldots,t_{n_{obs}}^\star} = \texttt{Marginalise}\left(\rho^{(\theta,t)|t_1^\star,\ldots,t_{n_{obs}}^\star},\, \{d_p+1, \ldots, d_p+d_s\}\right) \quad . $$

Simulations from the RMRP estimate of the posterior predictive density obtained using $\texttt{SimulateData}\left({}^{\square}f^{t|t_1^\star,\ldots,t_{n_{obs}}^\star}\right)$ can be used to check the Bayesian model ${}^{\square}f^{\theta|t_1^\star,\ldots,t_{n_{obs}}^\star}$ (O'Hagan and Forster 2004, chap. 8).

9.3.6 What if? analysis

The joint density estimate ${}^{\square}f^{(\theta,t)}$ (Section 9.3.2) can also be used to get additional insights into the relationships between the summary statistics representing the data and the parameters by applying `MRPSlice` (Algorithm 3.7) to obtain a number of conditional density estimates given specific values of θ.

Recall that $\rho^{(\theta,t)}$ is the root node of the RMRP joint density estimate ${}^{\square}f^{(\theta,t)}$ and $\boldsymbol{x}_\rho^{(\theta,t)} \in \mathbb{R}^{(d_p+d_s)}$ is the root box of ${}^{\square}f^{(\theta,t)}$. The coordinates of $\boldsymbol{x}_\rho^{(\theta,t)}$ are

$$ \Delta = \{1, \ldots, d_p, d_p+1, \ldots, d_p+d_s\} $$

and $\boldsymbol{x}_\rho^{(\theta,t)} = \boldsymbol{x}^{(\theta)} \times \boldsymbol{x}^{(t)}$ where $\boldsymbol{x}^{(\theta)}$ is a box representing the parameter space and $\boldsymbol{x}^{(t)}$ is a box representing the support of the summary statistics.

Given some specific value $\theta^\star$ for θ to be investigated, the subset of the coordinates of Δ used in the `MRPSlice` operation is $\Lambda = \{1, \ldots, d_p\}$ and the fixed subtuple point on which to slice is $\left({\theta^\star}_j^\dagger\right)_{j\in\Lambda}$ where ${\theta^\star}_j^\dagger = \theta_j^\star$, the j^{th} element in $\theta^\star$.

$\texttt{MRPSlice}(\rho^{(\theta,t)}, \{1, \ldots, d_p\}, \theta^{\star\dagger})$ gives the root node of an RMRP ${}^{\square}f^{|\theta^{\star\dagger}}$ representing an unnormalised estimate of $f\left(t \,|\, \theta^\star\right)$. Normalising ${}^{\square}f^{|\theta^{\star\dagger}}$ gives an RMRP ${}^{\square}f^{t|\theta^\star}$ representing the conditional density $f\left(t \,|\, \theta^\star\right)$. The root box of ${}^{\square}f^{t|\theta^\star}$ is $\boldsymbol{x}^{(t)}$, the box representing the support of the summary statistics from the simulated data.

An estimate $\hat{\mathcal{L}}(\theta^\star)$ of the point product likelihood of $\theta^\star$ given the observations can also be

calculated as the product of the pointwise images of each observation under the RMRP density estimate ${}^{\square}f^{t|\theta^\star}$ (see Section 3.6.2.4):

$$\hat{\mathcal{L}}(\theta^\star) = \prod_{i=1}^{n_{obs}} {}^{\square}f^{t|\theta^\star}(t_i^\star) \ , \quad \text{or, using log-likelihoods, } \hat{\ell}(\theta^\star) = \sum_{i=1}^{n_{obs}} \log\left({}^{\square}f^{t|\theta^\star}(t_i^\star)\right) \ .$$

Several different conditional density estimates and estimates of the point product likelihood can be compared using different values of $\theta^\star$. Once the joint density estimate ${}^{\square}f^{(\theta,t)}$ has been formed, the time taken to create each conditional density estimate is minimal and, again because of the RP tree structure, looking up ${}^{\square}f^{t|\theta^\star}(t_i^\star)$ for each summary statistic $t_i^\star$, $i = 1, \ldots, n_{obs}$, of the observed data is also very fast.

9.3.7 Combining AABC and RPABC

Sections 8.3 and 8.4.1 note that a final conditional density estimation step after the main AABC process is often used to try to adjust for the effect of the tolerance values $\{\varepsilon_j\}$ and improve the approximation to the target posterior. The AABC procedure itself may use relatively wide tolerance values. Beaumont et al. (2002) and Beaumont et al. (2009) used a form of local regression to accomplish the conditional density estimation step. If the AABC posterior density estimate is in the form of an RMRP, the operational capabilities of the RP structure can be exploited to obtain the conditional density using `MRPSlice` (Algorithm 3.7). This approach is a form of hybrid between AABC and RPABC and is referred to in this chapter as the AABC-RPABC method.

Let ${}^{\square}f^{\theta|\approx t^\star}$ denote the RMRP density estimate of $f(\theta \mid t^\star)$ obtained as the outcome of a AABC process for summary statistic $t^\star$. This estimate is based on the n values of θ' accepted in the final iteration. The acceptance step requires data x' to be simulated from $f(x \mid \theta')$ and the summary statistic t' of x' compared with $t^\star$ (see Section 8.4.1). Retaining the summary statistics of the accepted values of θ' and combining each θ' with its associated t' gives a sample of joint tuples $(\theta'^{(i)}, t'^{(i)}), \ldots, (\theta'^{(n)}, t'^{(n)})$ from an approximate posterior joint density $f((\theta,\, t) \mid \approx t^\star)$. This sample can be used to obtain an RMRP density estimate ${}^{\square}f^{(\theta,t)|\approx t^\star}$ of $f((\theta,\, t) \mid \approx t^\star)$. An estimate of the conditional density $f(\theta \mid t^\star)$ can then be obtained using `MRPSlice` by a similar process to that described in Section 9.3.3.

Let $\rho^{(\theta,t)|\approx t^\star}$ denote the root node of ${}^{\square}f^{(\theta,t)|\approx t^\star}$ and let $\boldsymbol{x}_\rho^{(\theta,t)|\approx t^\star} \in \mathbb{R}^{(d_p+d_s)}$ denote the root box of ${}^{\square}f^{(\theta,t)|\approx t^\star}$. The coordinates of $\boldsymbol{x}_\rho^{(\theta,t)|\approx t^\star}$ are $\Delta = \{1, \ldots, d_p, d_p+1, \ldots, d_p+d_s\}$.

The subset of the coordinates Δ used in the `MRPSlice` operation is $\Lambda = \{d_p{+}1, \ldots, d_p{+}d_s\}$, the coordinates of the joint sample tuples from the AABC process that represent the summary statistics. The fixed subtuple point on which to slice is $(t^{\star\dagger}_j)_{j\in\Lambda}$ where $t^{\star\dagger}_j = t^\star_{(j-d_p)}$, the $(j-d_p)^{\text{th}}$ element in $t^\star$.

`MRPSlice`$(\rho^{(\theta,t)|\approx t^\star}, \{d_p+1, \ldots, d_p+d_s\}, t^{\star\dagger})$ gives the root node of an RMRP ${}^{\square}f^{|t^{\star\dagger}}$ representing an unnormalised estimate of $f(\theta \mid t^\star)$. Normalising ${}^{\square}f^{|t^{\star\dagger}}$ gives an RMRP estimate

${}^{\square}f^{\theta|t^\star}$ of the posterior density $f(\theta \mid t^\star)$.

9.4 Toy model example in $\mathbb{R}^2$

This section uses a univariate mixture model to demonstrate RPABC in comparison with the AABC method of Beaumont et al. (2009) summarised in Section 8.4.1. The example is based on the toy model studied in Sisson et al. (2007) and Beaumont et al. (2009). Results from the AABC-RPABC hybrid method discussed in Section 9.3.7 are also shown.

9.4.1 The model and experimental setting

The aim was to make inferences about the location parameter $\mu \in \mathbb{R}$ in a univariate Normal mixture model. The data were realisations of a random variable $X \in \mathbb{R}$ such that

$$f(X = x \mid \mu) = \frac{1}{2}\varphi(x \mid \mu, 1) + \frac{1}{2}\varphi(x \mid \mu, \tfrac{1}{100}),$$

where $\varphi(x \mid \mu, \sigma^2)$ is the univariate Normal density with variance σ^2. The summary statistic of x was x itself. Fifty independent 'observed' values were simulated from the model using $\mu = 0$. The first 10 values are listed in Section L.1 of Appendix L.

9.4.2 Inference using AABC

This section describes the results obtained using an implementation of the AABC method of Beaumont et al. (2009) summarised in Section 8.4.1. The prior density $f(\mu)$ was the Uniform(-10, 10) density. The AABC process was used to obtain an estimate of the posterior density $f(\mu \mid x_i^\star)$ for each of the first 10 observations ($i = 1, \ldots, 10$) and for all these 10 observations summarised using the average observed value, denoted by $x^\star_{\cup 10}$. The AABC process was also used to obtain an estimate of the posterior density for all 50 observations by their average value, denoted $x^\star_{\cup 50}$.

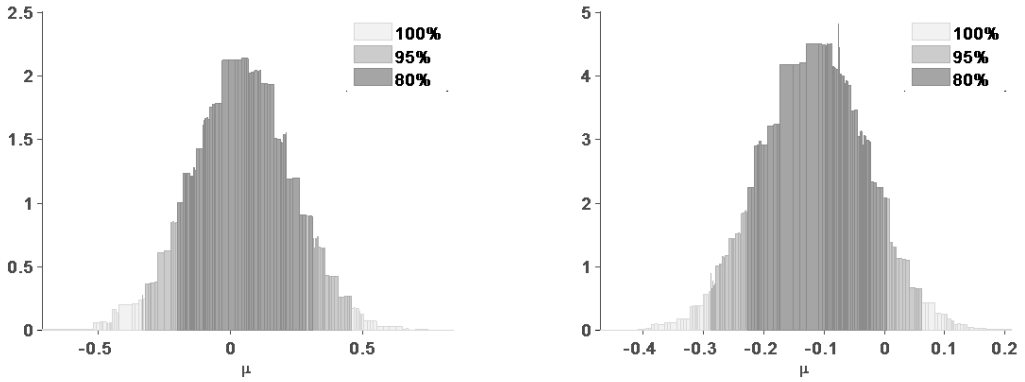
The number of simulations used within each iteration was $n = 10,000$ and four iterations were carried out. The tolerance level ε_1 for the first iteration was set so that 100,000 values of μ' were simulated and 10% of these closest to the observed data $x^\star$ were retained. In subsequent iterations the tolerance levels were set as a fraction of the tolerance used in the previous iteration: $\varepsilon_2 = 0.75\varepsilon_1$, $\varepsilon_3 = 0.75\varepsilon_2$, $\varepsilon_4 = 0.50\varepsilon_3$. This method of setting the tolerance levels is commonly used in AABC (Beaumont et al. (2009)) because appropriate values depend on the variability of the simulated data, which may be difficult to estimate in advance of the simulation process.

The total number of simulations from the model $f(x \mid \mu)$ required over the four iterations of each individual AABC process was over 600,000 for each of the runs described in this section. The simulations for each set of four iterations took about 40–50 minutes.

RMRP representations of the posterior densities were created using the MCMC process described in Chapter 6 with three chains. Each posterior density estimate was the average of 100 SRP samples collected, using a thin-out value of 1,000, from all three chains after convergence was diagnosed. It took only a few seconds to make each RMRP.

Figure 9.1(a) and 9.1(b) show the RMRP representations and highest posterior density regions for the AABC estimates of $f\left(\mu \mid x^{\star}_{\cup 10}\right)$ and $f\left(\mu \mid x^{\star}_{\cup 50}\right)$, respectively. The highest posterior density regions were found using the `CoverageRegion` operation (Algorithm 3.11). The 95% highest posterior density region for $f\left(\mu \mid x^{\star}_{\cup 10}\right)$ was [-0.33, 0.45]. The estimated 95% highest posterior density region for $f\left(\mu \mid x^{\star}_{\cup 50}\right)$ was [-0.29, 0.06]. The nature of the model and the nature of the AABC algorithm together mean that the mode of the posterior density estimate is close to the single observation used in the AABC process: $x^{\star}_{\cup 10} = 0.042$ for Figure 9.1(a), and $x^{\star}_{\cup 10} = -0.118$ for Figure 9.1(b). This is discussed in more detail later, in Section 9.6.2, in relation to another example.

Figure 9.2 shows the RMRP estimates of the posteriors $f\left(\mu \mid x^{\star}_{1}\right)$ and $f\left(\mu \mid x^{\star}_{8}\right)$ obtained from AABC processes for the individual observations $x^{\star}_{1}$ and $x^{\star}_{8}$, respectively.



(a) Posterior density estimate given $x^{\star}_{\cup 10}$.

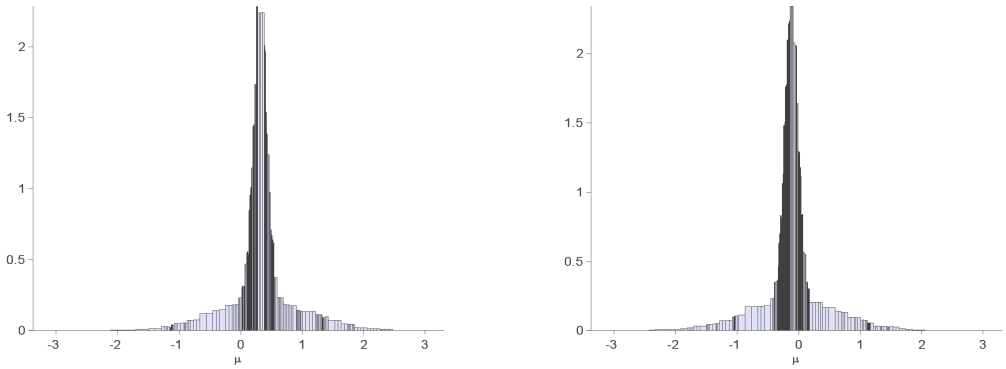
(b) Posterior density estimate given $x^{\star}_{\cup 50}$.

Figure 9.1: AABC posterior density estimates.

9.4.3 Inference using RPABC

This section discusses the results obtained using an implementation of the RPABC process described in Section 9.3. The RPABC process was used to obtain an estimate of the posterior density $f\left(\mu \mid x^{\star}_{1}, \ldots, x^{\star}_{10}\right)$ (the posterior given the data for the first 10 observations as independent samples) and also for the posterior densities $f\left(\mu \mid x^{\star}_{1}, \ldots, x^{\star}_{30}\right)$ and $f\left(\mu \mid x^{\star}_{1}, \ldots, x^{\star}_{50}\right)$, to demonstrate the effect of more independent observations on the posterior.

The prior density $f(\mu)$ was the Uniform(-8, 8) density. The prior support was calculated using the heuristic routine discussed in Section 9.3.1 and described in detail in Appendix K. The preliminary prior support used in the routine was [-10, 10].



(a) Posterior density estimate given $x_1^\star$.

(b) Posterior density estimate given $x_8^\star$.

Figure 9.2: Posterior density estimates using individual observations.

9.4.3.1 Estimating the joint density

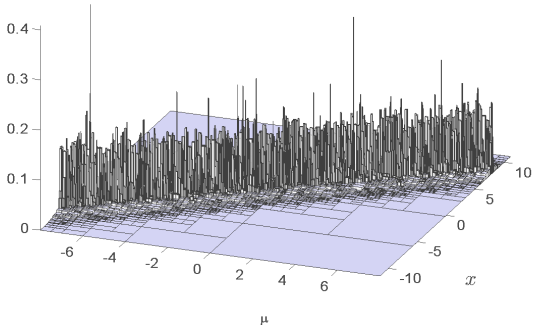


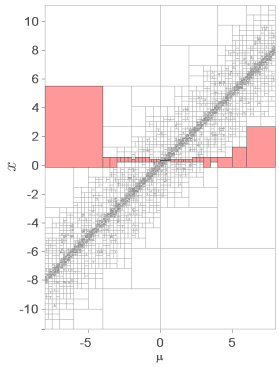
Figure 9.3: Estimating the joint density $f(\mu, x)$.

The number of simulations used to estimate the joint density $f(\mu, x)$ was $n = 100,000$. Only one joint density estimate was needed to be able to obtain all of the posteriors listed above. The MCMC process used three chains. The joint density estimate was the average of 100 samples collected, using a thin-out value of 1,000, from all three chains after convergence was diagnosed. Figure 9.3 shows the RMRP joint density estimate.

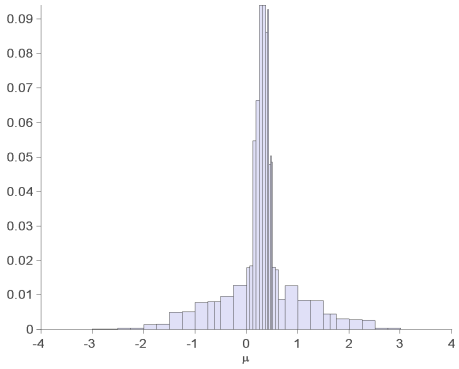
9.4.3.2 Estimating the posterior density

Figure 9.4 illustrates slicing to obtain the posterior, as described in Section 9.3.3, using $x_1^\star = 0.313$ and $x_8^\star = -0.119$, the largest and smallest, respectively, of the first 10 observations. Figure 9.4(a) shows the elements of the partition of the support of the RMRP joint density

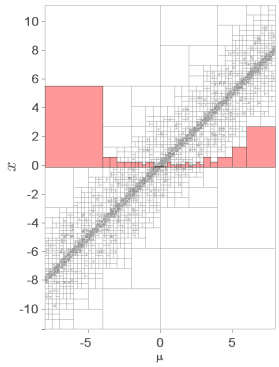
estimate through which the slice on $x_1^\star$ passes. Figure 9.4(b) shows the RMRP of the resulting RMRP estimate of the unnormalised conditional function $f(\mu \mid x_1^\star)$. Figures 9.4(c) and 9.4(d) show equivalent plots for a slice on $x_8^\star$.



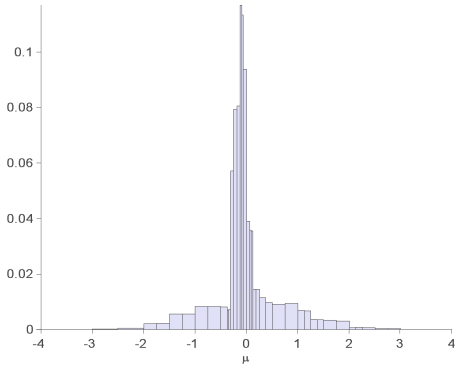
(a) Elements of the partition though which the slice on $x_1^\star$ passes.



(b) Unnormalised slice on $x_1^\star$.



(c) Elements of the partition though which the slice on $x_8^\star$ passes.

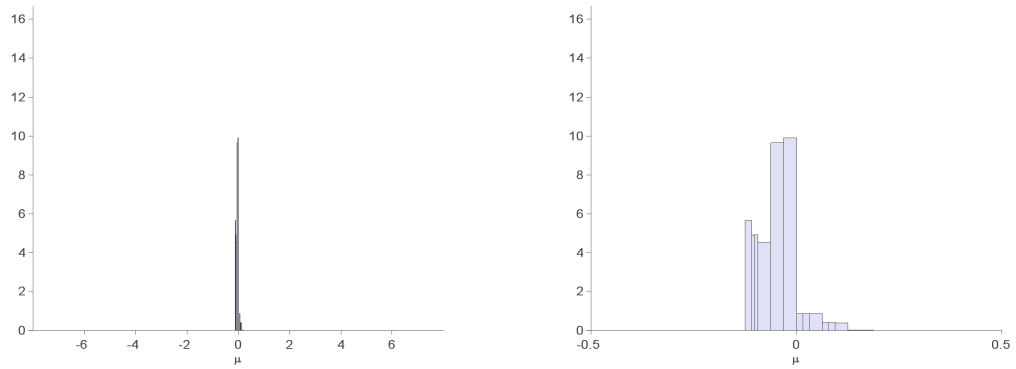


(d) Unnormalised slice on $x_8^\star$.

Figure 9.4: Slicing the joint density estimate.

Figure 9.5 shows the RPABC RMRP estimate of $f(\mu \mid x_1^\star, \ldots, x_{10}^\star)$, the posterior given the data for the first 10 observations, obtained by multiplying the slices on $x_1^\star, \ldots, x_{10}^\star$ and adjusting for $f(\mu)$ as described in Section 9.3.3. Figure 9.5(a) shows the posterior estimate relative to the support of the prior. Figure 9.5(b) shows the posterior estimate and highest posterior density regions on a rescaled horizontal axis so as to be see the detail more clearly. The scale is now very much smaller than that used to show the results of the AABC method using $x_{\bigcup 10}^\star$ and $x_{\bigcup 50}^\star$ (Figures 9.1(a) and 9.1(b), respectively). Exploiting the product likelihood structure through the multiplications of slices, such as those shown in Figures 9.4(b) and 9.4(d), resulted in very concentrated posterior density estimates.

Figure 9.6 illustrates the effect of increasing the number of observations used to obtain

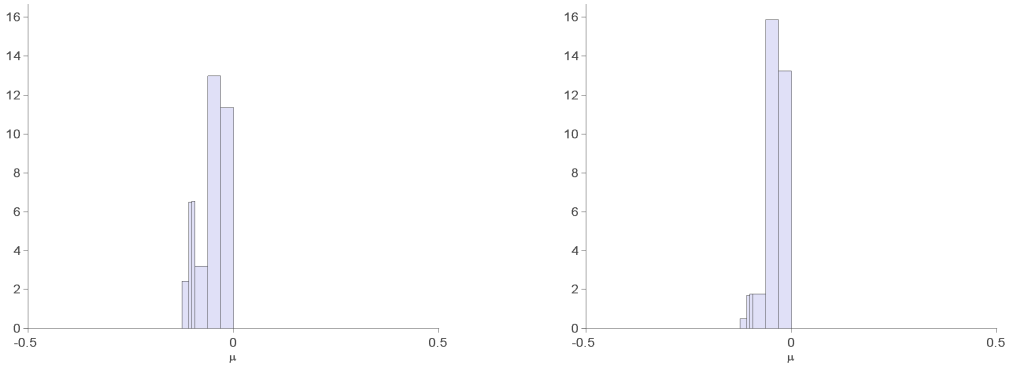


(a) On the scale of the prior support.

(b) Figure 9.5(a) rescaled.

Figure 9.5: Estimating $f(\mu \mid x_1^\star, \ldots, x_{10}^\star)$.

the posterior density estimate, showing RPABC RMRP estimates of the posterior density and highest posterior density regions, on the same μ-scale as Figure 9.5(b), using the data for 30 and 50 observations.



(a) Estimating $f(\mu \mid x_1^\star, \ldots, x_{30}^\star)$.

(b) Estimating $f(\mu \mid x_1^\star, \ldots, x_{50}^\star)$.

Figure 9.6: Posterior estimates with $n_{obs} = 30$ and $n_{obs} = 50$.

The 95% highest posterior density region from the estimate of $f(\mu \mid x_1^\star, \ldots, x_{10}^\star)$ was [-0.12, 0.06]. The 95% highest posterior density region for both $f(\mu \mid x_1^\star, \ldots, x_{30}^\star)$ and $f(\mu \mid x_1^\star, \ldots, x_{50}^\star)$ was [-0.11 , 0.0].

9.4.3.3 What if? analysis

Figure 9.7(a) shows an RMRP estimate ${}^{\square}f^{x|\mu^\star=0}$ of $f(x \mid \mu^\star = 0)$, the density of the data conditional on the value of μ used to simulate the observed data, obtained from the joint density estimate using `MRPSlice` and `Normalise` as described in Section 9.3.6. The observed data is superimposed on the support of the density estimate. The path of the slice on $\mu^\star = 0$

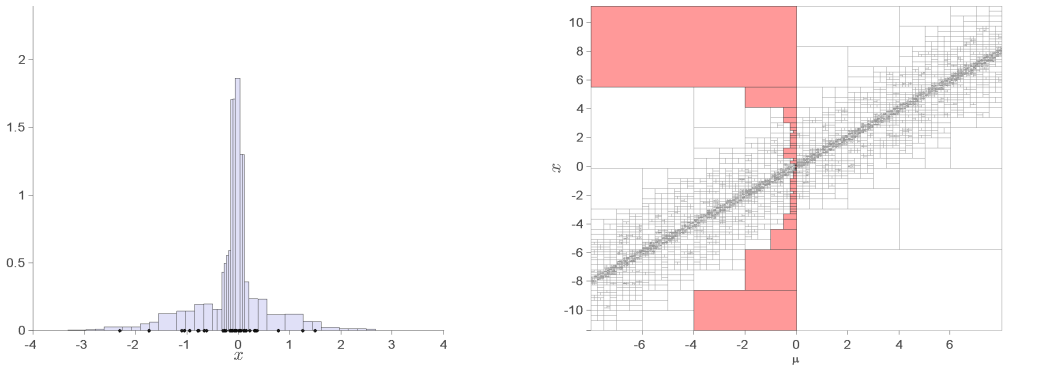
through the partition of the joint density RMRP is shown in Figure 9.7(b).

Figure 9.8(a) shows an RMRP estimate of $f\left(x \,|\, \mu^\star = 0.25\right)$, the density of the data conditional on $\mu = 0.25$, again with the values of the observed data superimposed. Similar estimates for $\mu^\star = 0.50$, $\mu^\star = 1.00$, and $\mu^\star = 1.50$ are shown in Figures 9.8(b), 9.8(c),and 9.8(d), respectively. Each of these conditional density estimates can be obtained very quickly from the single joint density created at the start of the RPABC process.

Table 9.1 shows the point log-likelihood estimates calculated using the product of the images, under each of the five RMRP conditional density estimates, of the observed values for 10, 20, 30, 40, and 50 observations, as described in Section 9.3.6. For example, the estimate of the log-likelihood of $\mu^\star = 0.25$ using the first 10 observations is

$$\hat{\ell}(0.25) = \sum_{i=1}^{10} \log\left({}^{\square}f^{x|0.12}(x_i^\star)\right)$$

where ${}^{\square}f^{x|0.25}$ is illustrated in Figure 9.8(a). This toy model, of course, does not present a particularly demanding inference problem and it is not surprising that the log-likelihood is higher for $\mu^\star = 0$, the value of μ used to simulate the observed data used for this example, than for the other values of $\mu^\star$ in each row of Table 9.1).



(a) ${}^{\square}f^{x|\mu^\star=0}$.

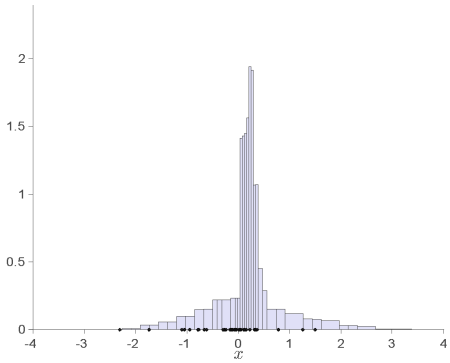
(b) Path of the slice through the joint partition.

Figure 9.7: RMRP estimate of $f\left(x \,|\, \mu^\star = 0\right)$.

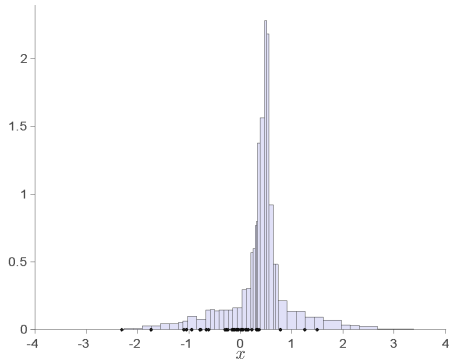
9.4.3.4 Estimating the posterior predictive density

Simulation from ${}^{\square}f^{\mu|x_1^\star,\ldots,x_{n_{obs}}^\star}$ using `SimulateData` (Algorithm 3.12) and `Marginalise` (Algorithm 3.10) can be used to create an estimate of the posterior predictive density $f\left(x \,|\, x_1^\star, \ldots, x_{n_{obs}}^\star\right)$, as described in Section 9.3.5.

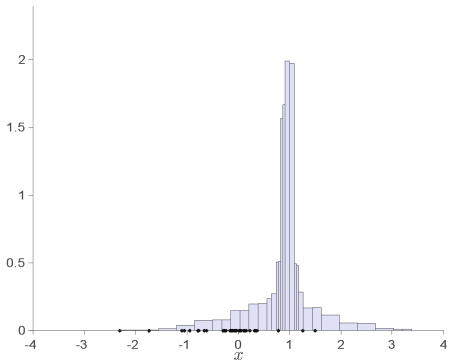
Figure 9.9(a) shows the RMRP estimate and coverage regions for the posterior predictive density given the values of the first 10 observations. These observations are overlaid as points



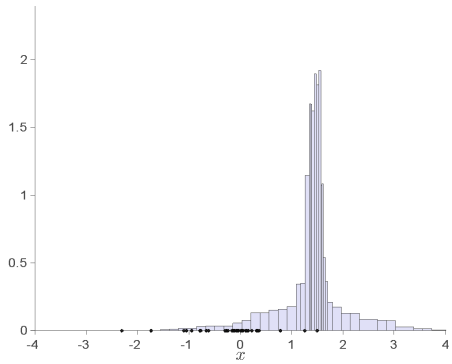
(a) RMRP estimate of $f\left(x \mid \mu^{\star} = 0.25\right)$.



(b) RMRP estimate of $f\left(x \mid \mu^{\star} = 0.50\right)$.



(c) RMRP estimate of $f\left(x \mid \mu^{\star} = 1.00\right)$.



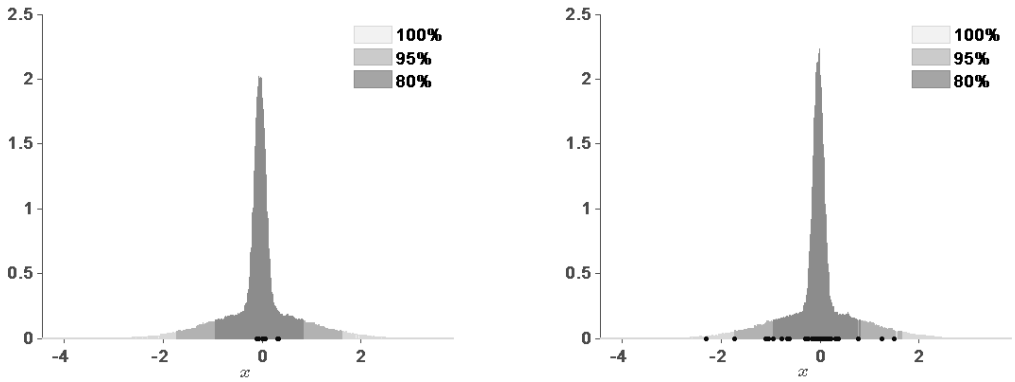
(d) RMRP estimate of $f\left(x \mid \mu^{\star} = 1.50\right)$.

Figure 9.8: Estimating $f\left(x \mid \mu^{\star}\right)$ using slices of the joint density estimate.

Table 9.1: Point log-likelihoods for various values of $\mu^{\star}$.

$\mu^{\star}$	0.0	0.25	0.5	1.0	1.5
$\hat{\ell}_{10}(\mu^{\star})$	0.6	−6.1	−13.2	−18.5	−26.3
$\hat{\ell}_{20}(\mu^{\star})$	−8.7	−23.1	−35.4	−45.5	−62.8
$\hat{\ell}_{30}(\mu^{\star})$	−14.5	−38.5	−57.3	−70.2	−96.5
$\hat{\ell}_{40}(\mu^{\star})$	−20.7	−49.0	−72.5	−90.0	−124.5
$\hat{\ell}_{50}(\mu^{\star})$	−29.6	−54.8	−85.3	−109.6	−147.9

on the support of the density. Figure 9.9(b) shows the RMRP estimate and coverage regions of the posterior predictive density given all 50 observations with these observed values overlaid.



(a) RMRP estimate of $f\left(x \mid x_1^\star, \ldots, x_{10}^\star\right)$.

(b) RMRP estimate of $f\left(x \mid x_1^\star, \ldots, x_{50}^\star\right)$.

Figure 9.9: Estimating the posterior predictive density, with the observed data overlaid.

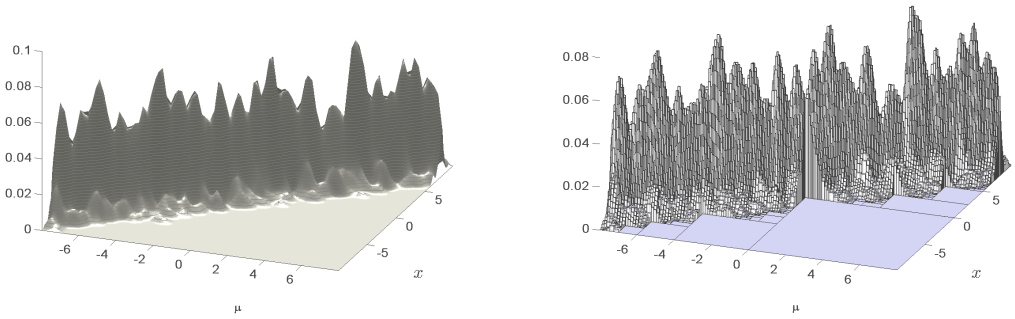
9.4.3.5 Approximating a KDE of the joint density

An alternative to the SRP MCMC method for forming the RMRP estimate of the joint density is to use a kernel density estimator to form a KDE and then approximate this with an RMRP as described in Chapter 7. A smaller sample of data can be used, which would be a particular advantage when simulating data from the model is much more computationally expensive than it is for this Normal mixture example.

Figure 9.10(a) shows a KDE of the joint density obtained using a sample size $n_K = 2,000$ and the multivariate MCMC bandwidth kernel density estimator of Zhang et al. (2006) discussed in Chapter 7. Figure 9.10(b) shows an RMRP approximation to the KDE obtained using the method described in Section 7.3 and approximation tolerance $\overline{\psi} = 0.00001$.

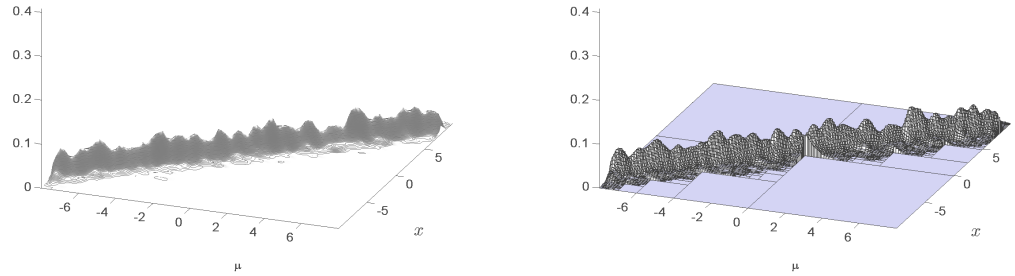
Figure 9.11(a) shows the KDE of Figure 9.10(a) with the same vertical scale as the RMRP estimate obtained using the MCMC SRP process shown in Figure 9.3. Figure 9.11(a) shows the RMRP approximation to the KDE similarly rescaled. The rapid rate of change in the density around the diagonal $x = \mu$ is very challenging for the diagonal bandwidth matrix used by the kernel density estimator, resulting in oversmoothing around this modal ridge. The RMRP from the SRP MCMC process errs in the other direction: it is much more locally-adaptive and, as Figure 9.3 shows, is undersmoothed. It is possible that using a larger data sample for the kernel density estimate would have helped somewhat but the time required to form the KDE then increases massively because of the MCMC bandwidth selection process. No tests with larger sample sizes were carried out because it seemed likely that some form of locally-adaptive bandwidth estimator would be required to cope properly with this data.

With a suitable kernel density estimator the rest of the analysis would proceed exactly as



(a) KDE for $f(\mu, x)$. (b) RMRP approximation to the KDE.

Figure 9.10: Using a KDE to approximate the joint density.



(a) Figure 9.10(a) rescaled. (b) Figure 9.10(b) rescaled.

Figure 9.11: KDE and approximation to KDE rescaled.

above, obtaining the posterior density estimate, etc., by operating on the RMRP approximation to the KDE instead of the RMRP formed using the SRP MCMC method.

9.4.3.6 RPABC timings

The initial step to determine a suitable prior support described in Section 9.3.1 took less than a second. Generating $n = 100,000$ joint samples from the prior $f(\mu)$ and the model $f(x \mid \mu)$ also took less than a second. Forming the RMRP estimate of the joint density using the MCMC process took about 12–15 minutes. The operations on the joint density estimate to obtain the posterior densities and the conditional density functions on various parameters of interest took less than 5 seconds. Estimating each of the posterior predictive densities took 7–8 hours. These, and all the other times given in this chapter, were recorded while other processes were running on the same machine and can only be taken as giving a general indication of the

running time required by the process. Where a range of times is given this is based on the range of times experienced for these examples when testing and developing the RPABC method.

9.4.4 Combining AABC and RPABC

This section describes the results obtained using the hybrid AABC-RPABC method described in Section 9.3.7. An RMRP density estimate ${}^{\square}f^{\mu|\approx x^\star_{\cup 10}}$ of the approximate posterior $f\left(\mu \,|\, \approx x^\star_{\cup 10}\right)$ was created using just one iteration in the main AABC process but with a larger number ($n = 100,000$) of simulated samples of μ' drawn from the approximate posterior than in Section 9.4.4. The number of initial draws from the prior was 1,000,000 and ε_1 was set so that 10% of these initial draws with $x' \sim f\left(x \,|\, \mu'\right)$ closest to $x^\star_{\cup 10}$ were retained. These, together with their associated values of x', formed the sample on which to base the RMRP estimate of the posterior joint density $f\left((\mu,\, x) \,|\, \approx x^\star_{\cup 10}\right) = f\left(x \,|\, \mu\right) f\left(\mu \,|\, \approx x^\star_{\cup 10}\right)$.

Figure 9.12(a) shows the RMRP density estimate of this posterior joint density. Figure 9.12(b) shows, shaded, the elements of this RMRP's partition through which the slice on $x^\star_{\cup 10}$ passes. Figure 9.13(a) shows the RMRP density estimate of $f\left(\mu \,|\, \approx x^\star_{\cup 10}\right)$ obtained as a result of applying the AABC process with just one iteration. Figure 9.13(b) shows the refined RMRP density estimate of $f\left(\mu \,|\, x^\star_{\cup 10}\right)$ after slicing the joint estimate shown in Figure 9.12(a) on $x^\star_{\cup 10} = 0.042$.

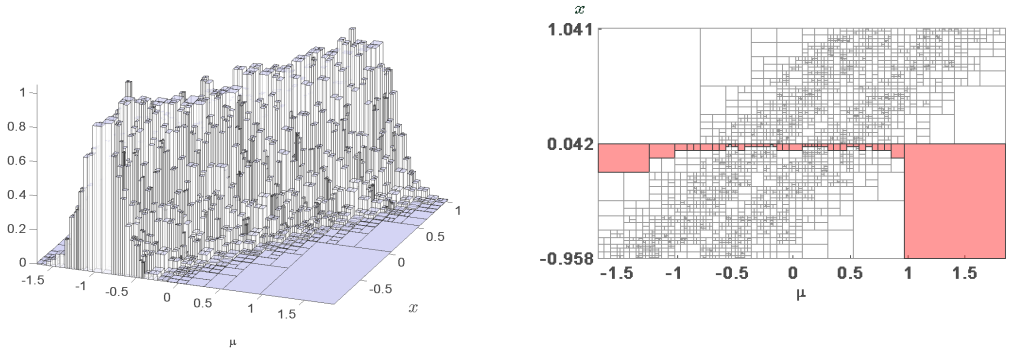
The estimate of $f\left(\mu \,|\, \approx x^\star_{\cup 10}\right)$ shown in Figure 9.13(a) has a much wider support than the posterior estimate shown in Figure 9.1(a), because just one iteration of the AABC process gives a much looser approximation. The 95% highest posterior density region for the refined estimate of the posterior $f\left(\mu \,|\, x^\star_{\cup 10}\right)$ shown in Figure 9.13(b) is [-0.48, 0.46]. This is a little wider than the 95% highest posterior density region obtained from the estimate using the full four AABC iterations and the first 10 observations shown in Figure 9.1(a) ([-0.33, 0.45]) and wider than the 95% highest posterior density region obtained from the RPABC RMRP estimate of $f\left(\mu \,|\, x^\star_1, \ldots, x^\star_{10}\right)$ in Figure 9.5 ([-0.12, 0.06]) .

The time taken to obtain the estimate shown in Figure 9.13(b) was about a minute (30 seconds to obtain the simulation samples and 30 seconds to crate the required RMRPs), compared with 40–50 minutes for Figure 9.1(a) and about 15 minutes to obtain RPABC estimates of $f\left(\mu \,|\, x^\star_1, \ldots, x^\star_{10}\right)$, $f\left(\mu \,|\, x^\star_1, \ldots, x^\star_{30}\right)$, and $f\left(\mu \,|\, x^\star_1, \ldots, x^\star_{50}\right)$ (Figures 9.5 and 9.6).

A full discussion of the advantages and disadvantages of the AABC-RPABC method in relation to RPABC and AABC is deferred until Section9.8. In this example the method provided a much more efficient means of obtaining a result similar to that given by a full AABC process, but this was not the case for all of the examples that follow in this chapter.

9.5 Toy model example in $\mathbb{R}^4$

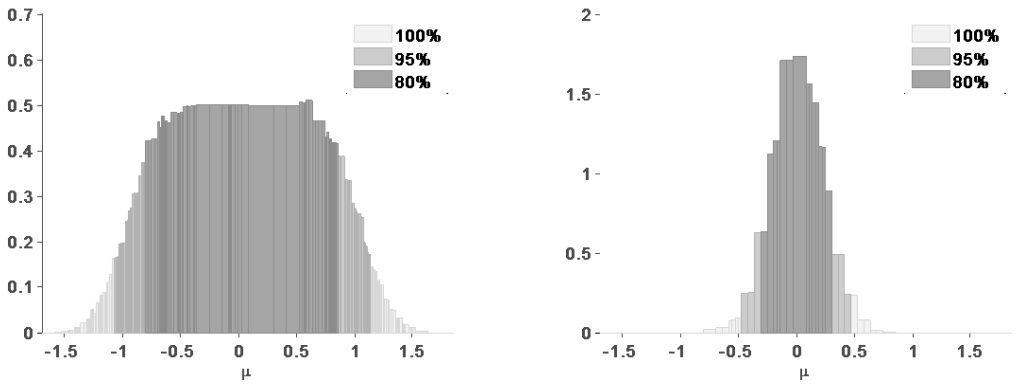
This section extends toy model to four dimensions.



(a) The joint density estimate.

(b) Elements of the partition of the joint density estimate through which the slice passes.

Figure 9.12: Slicing the joint density estimate on $x^{\star}_{\cup 10}$.



(a) Single AABC iteration estimate.

(b) Refined estimate after slicing.

Figure 9.13: Refining the posterior density estimate using `MRPSlice`.

9.5.1 The model and experimental setting

The aim was to make inferences about the location parameter $\mu \in \mathbb{R}^2$ in a bivariate Normal mixture model. The data were realisations of a random variable $X \in \mathbb{R}^2$ such that

$$f(X = x \,|\, \mu, \Sigma_a, \Sigma_b) = \frac{1}{2}\varphi(x \,|\, \mu, \Sigma_a) + \frac{1}{2}\varphi(x \,|\, \mu, \Sigma_b),$$

where $\varphi(x \,|\, \mu, \Sigma)$ is the bivariate Normal density with 2×2 variance-covariance matrix Σ, and

$$\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \quad \Sigma_a = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \Sigma_b = \begin{pmatrix} \frac{1}{100} & 0 \\ 0 & \frac{1}{100} \end{pmatrix}.$$

The summary statistic of $x = (x_1, x_2)$ was x itself. Fifty independent 'observed' values were simulated from the model using $\mu_1 = \mu_2 = 0$. Figure 9.14 shows a plot of these $n_{obs} = 50$

simulations. The first 10 values are listed in Section L.1 of Appendix L.

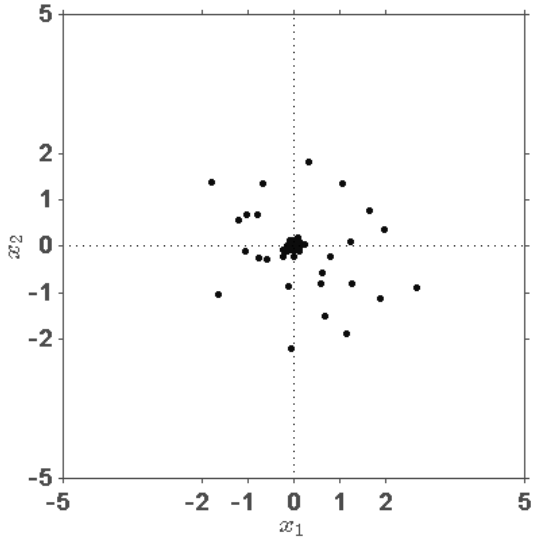


Figure 9.14: $n_{obs} = 50$ observed values.

9.5.2 Inference using AABC

This section describes the results obtained using an implementation of the AABC method of Beaumont et al. (2009) summarised in Section 8.4.1. The prior density for $f(\mu)$ was the two dimensional Uniform density on $[-10, 10]^2$.

The AABC process was used to obtain an estimate of the posterior density $f(\mu \mid x_i^\star)$ for each of the first 10 observations ($i = 1, \ldots, 10$) and for all these 10 observations summarised as the average observed value, denoted by $x^\star_{\cup 10}$. The AABC process was also used to obtain an estimate of the posterior density for all 50 observations summarised by their average, denoted by $x^\star_{\cup 50}$.

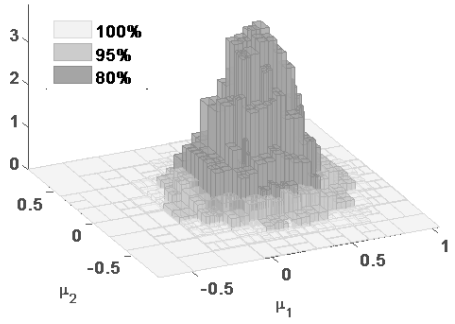
The number of simulations used within each iteration was $n = 10,000$ and four iterations were carried out to obtain each posterior density estimate. The tolerance level ε_1 for the first iteration was set so that 100,000 values of μ' were simulated and 10% of these closest to the observed data were retained. In subsequent iterations the tolerance levels were set as a fraction of the tolerance used in the previous iteration: $\varepsilon_2 = 0.75\varepsilon_1$, $\varepsilon_3 = 0.75\varepsilon_2$, $\varepsilon_4 = 0.50\varepsilon_3$.

In total, more than a million simulations from the model, taking 50–60 minutes to generate, were required over the four iterations of each individual AABC process described in this section.

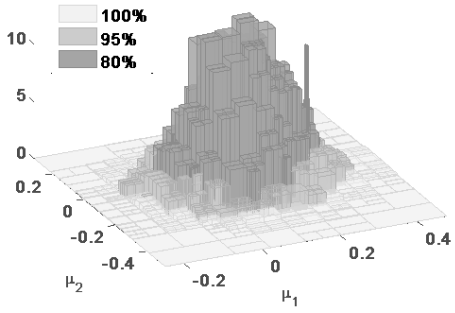
RMRP representations of the posterior densities were created using the MCMC process described in Chapter 6. The MCMC process used three chains. Each posterior density estimate was the average of 100 samples collected, using a thin-out value of 1,000, from all three chains after convergence was diagnosed.

Figures 9.15(a) and 9.15(b) show the RMRP density estimates, with highest posterior density regions, of the AABC posteriors $f\left(\mu \,|\, x^{\star}_{\cup 10}\right)$ and $f\left(\mu \,|\, x^{\star}_{\cup 50}\right)$, respectively. The highest posterior density regions are shown in plan view in Figures 9.15(c) and 9.15(d).

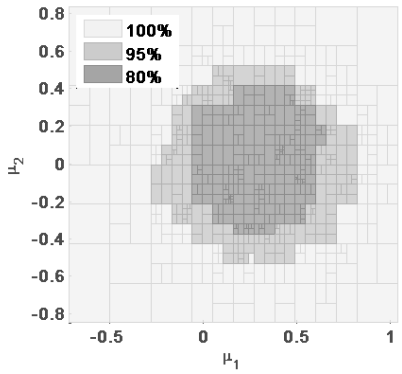
In this example the AABC processes using the individual observations can each give a very different posterior density estimate, because of the variability of the observations. Figure 9.16 shows the RMRP estimates of the posteriors $f\left(\mu \,|\, x^{\star}_{1}\right)$ and $f\left(\mu \,|\, x^{\star}_{8}\right)$ obtained from AABC processes for the first and eighth individual observations, respectively.



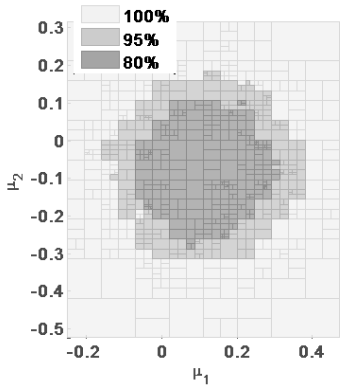
(a) Posterior density estimate given $x^{\star}_{\cup 10}$.



(b) Posterior density estimate given $x^{\star}_{\cup 50}$.



(c) Plan view of Figure 9.15(a).



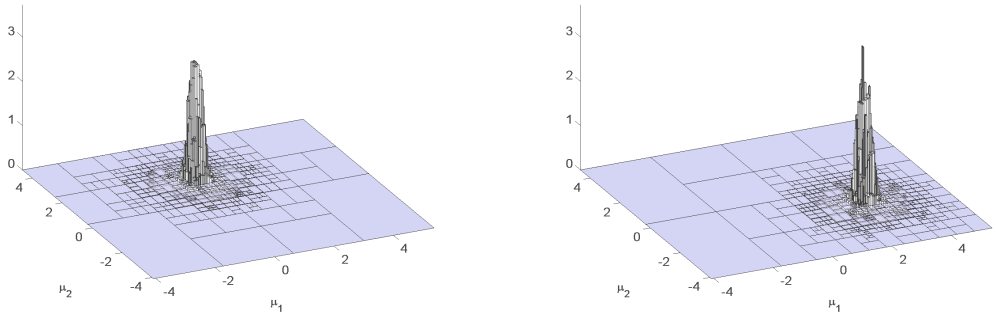
(d) Plan view of Figure 9.15(b).

Figure 9.15: AABC posterior density estimates.

9.5.3 Inference using RPABC

This section discusses the results obtained using an implementation of the RPABC process described in Section 9.3. The RPABC process was used to obtain an estimate of the posterior density given the first 10 independent observations, $f\left(\mu \,|\, x^{\star}_{1}, \ldots, x^{\star}_{10}\right)$, and also for the posterior densities $f\left(\mu \,|\, x^{\star}_{1}, \ldots, x^{\star}_{30}\right)$ and $f\left(\mu \,|\, x^{\star}_{1}, \ldots, x^{\star}_{50}\right)$.

The prior density for $f(\mu)$ was the two dimensional Uniform density on $[-10, 10]^2$. The



(a) Posterior density estimate given $x_1^\star$.

(b) Posterior density estimate given $x_8^\star$.

Figure 9.16: Posterior density estimates using individual observation.

heuristic routine discussed in Section 9.3.1 and described in detail in Appendix K was used with preliminary prior support $[-10, 10]^2$ and did not result in any reduction in this prior support.

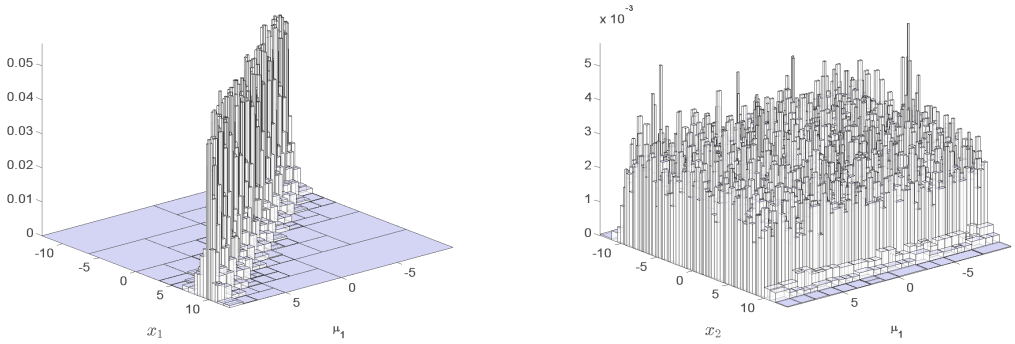
9.5.3.1 Estimating the joint density

The number of simulations used to estimate the joint density $f(\mu, x)$ was $n = 100,000$. The joint density estimate was the average of 100 samples collected, using a thin-out value of 1,000, from all three chains after convergence was diagnosed.

The four-dimensional RMRP estimate of the joint density can easily be marginalised to give two-dimensional estimates of the marginal density on each parameter and data-element pair. These marginal density estimates are displayed in Figure 9.17. The strong relationship between μ_1 and x_1 and absence of relationship between μ_1 and x_2 that would be expected from the model can be seen clearly. As would also be expected from the model, the RMRP that results from marginalising on (μ_2, x_1) is almost identical to Figure 9.17(b) and the RMRP that results from marginalising on (μ_2, x_2) is almost identical to Figure 9.17(a). Plots of these other marginal RMRPs are therefore not shown.

9.5.3.2 Estimating the posterior density

Figure 9.18 illustrates slicing to obtain the posterior, as described in Section 9.3.3, using $x_1^\star = (0.313, 1.813)$ and $x_8^\star = (2.671, -0.885)$. Figure 9.18(a) shows, shaded, the elements of the partition of the RMRP joint density estimate through which the slice on $x_1^\star$ passes using the marginal RMRP on the coordinates for μ_1 and x_2. Figure 9.18(b) shows a similar plot for the path of this slice with respect to the marginal density on the coordinates for μ_2 and x_2. Figures 9.18(c) and 9.18(d) similarly show the elements of the partition of the RMRP joint density estimate through which the slice on $x_8^\star$ passes using the marginal RMRPs



(a) Marginalisation on μ_1 and x_1.

(b) Marginalisation on μ_1 and x_2.

Figure 9.17: Marginals of the estimate of the joint density $f(\mu, x)$.

on the coordinates for μ_1,x_1 (Figure 9.18(c)) and μ_1, x_1 (Figure 9.18(d)). Like Figure 9.4, Figure 9.18 illustrates the subtle differences between the elements in the joint partition through which different slices may pass.

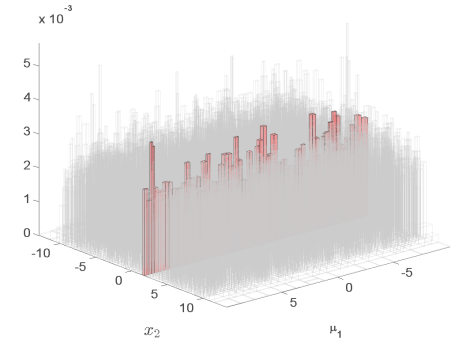
Figures 9.19(a) and 9.19(b) show the RMRPs of the unnormalised conditional functions $f(\mu \mid x_1^\star)$ and $f(\mu \mid x_8^\star)$ formed from these `MRPSlice` operations on the joint density estimate, showing how the apparently minor differences in the paths of the slices noted above results much more obvious differences between the actual conditional functions themselves.

The estimates of the posterior densities given multiple independent observations were obtained by multiplying the slices for the individual observations and adjusting for $f(\mu)$ using Equation (9.3). The 2-dimensional RMRP estimate of $f(\mu)$ used in the calculation was obtained by using `Marginalise` on the joint density estimate as described in Section 9.3.3.

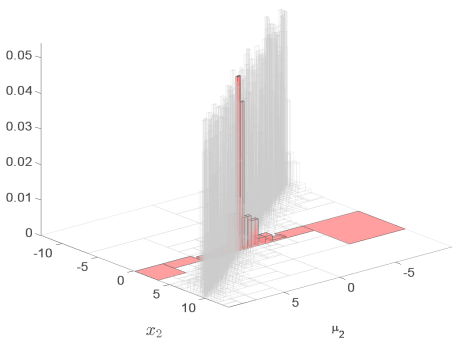
Figure 9.19 shows the RPABC RMRP estimate of $f(\mu \mid x_1^\star, \ldots, x_{10}^\star)$, Figure 9.19(c) shows the posterior estimate relative to the support of the prior. Figure 9.19(d) shows a closeup of the highest posterior density regions in plan view.

Figure 9.20 illustrates the effect of increasing the number of observations used to obtain the RPABC posterior density estimate, showing closeups of the highest posterior density regions of RPABC RMRP estimates of the posterior density using the data for 30 and 50 observations. The 95% highest posterior density regions from the estimates of $f(\mu \mid x_1^\star, \ldots, x_{30}^\star)$ and $f(\mu \mid x_1^\star, \ldots, x_{50}^\star)$ were both the single boxes $[0.00, 0.31] \times [-0.63, 0.00]$ in the RMRP partition.

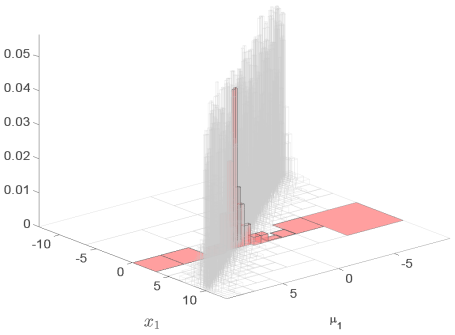
Almost all of the density of the final posterior estimates obtained in this RPABC process was concentrated on a few elements in the partition, giving very low resolution estimates of the highest posterior density regions. A very much finer partition would be needed in the joint density estimate (Figure 9.17) to be able to obtain smoother RPABC posterior density estimates. Using a larger number of simulation samples to create the joint density estimate would have helped to achieve this, but would have also greatly increased the time taken to



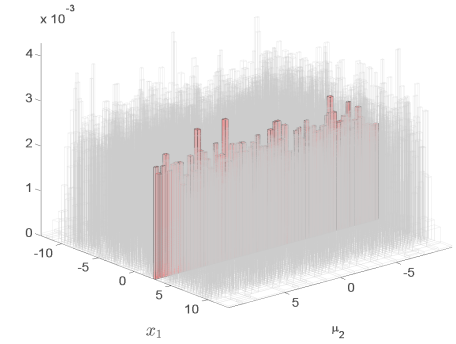
(a) Slice on $x_1^\star$, marginal variables (μ_1, x_2).



(b) Slice on $x_1^\star$, marginal variables (μ_2, x_2).

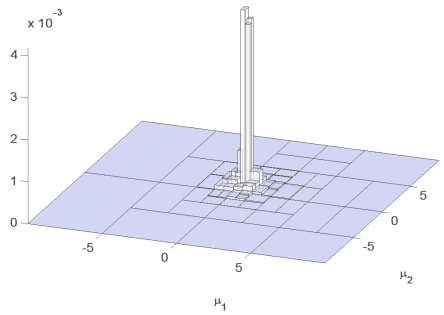


(c) Slice on $x_8^\star$, marginal variables (μ_1, x_1).

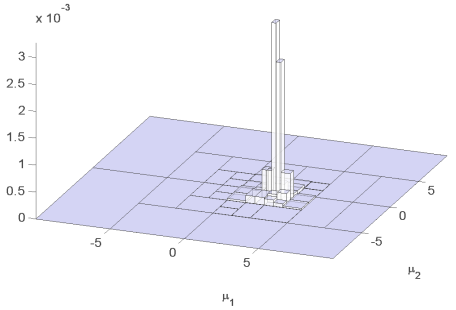


(d) Slice on $x_8^\star$, marginal variables (μ_2, x_1).

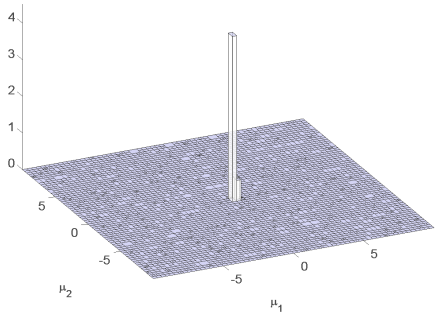
Figure 9.18: Slicing the joint density estimate.



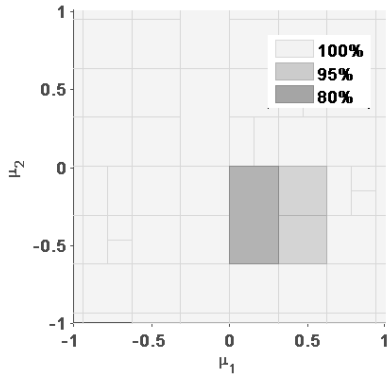
(a) Slice on $x_1^\star$.



(b) Slice on $x_8^\star$.



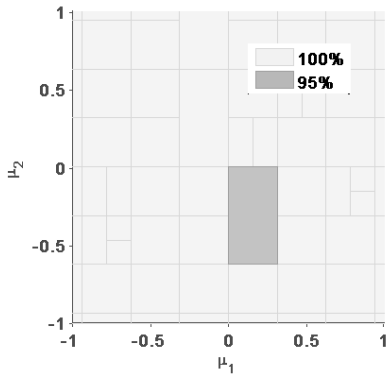
(c) On the scale of the prior support.



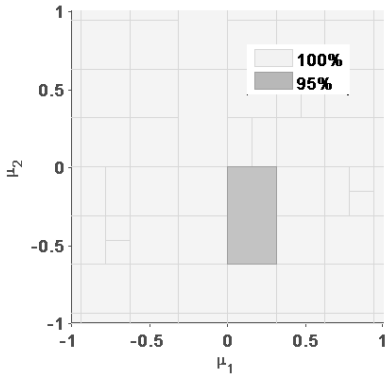
(d) Rescaled, highest posterior density regions.

Figure 9.19: Estimating $f\left(\mu \,|\, x_1^\star, \ldots, x_{10}^\star\right)$.

obtain the RMRP joint density estimate using the SRP MCMC method of Chapter 6.



(a) ${}^{\square}f^{\mu|x_1^\star,\ldots,x_{30}^\star}$.

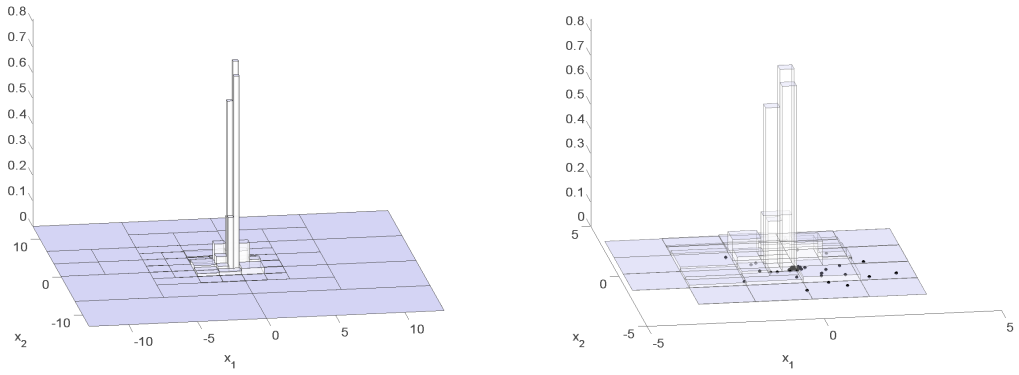


(b) ${}^{\square}f^{\mu|x_1^\star,\ldots,x_{50}^\star}$.

Figure 9.20: Posterior estimates with $n_{obs} = 30$ and $n_{obs} = 50$.

9.5.3.3 What if? analysis

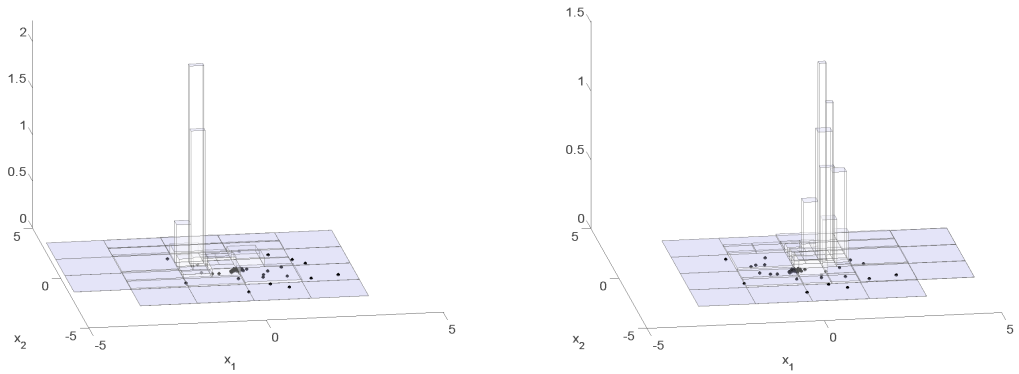
Figure 9.21 shows an RMRP estimate of $f\left(x \,|\, \mu_1 = 0,\, \mu_2 =\right)$, the density of the data conditional on the value of μ used to simulate the observed data. This was obtained from the joint density estimate using `MRPSlice` and `Normalise` as described in Section 9.3.6. Figure 9.21(a) is on the scale of the joint density. Figure 9.21(b) shows a closeup and excludes the boxes in the partition of the estimate that do not fit within the axis limits. The observed data is overlaid on the support of the density estimate. Figure 9.22 shows similar closeups of the RMRP estimates of the conditional density functions $f\left(x \,|\, \mu^\star\right)$ for $\mu^\star = (-1, 1)$ and $\mu^\star = (1, 1)$ also obtained by using `MRPSlice` as in Section 9.3.6.



(a) On the scale of the joint density.

(b) Closeup, with observed values.

Figure 9.21: Estimating $f(x \mid \mu^\star = (0,0))$.



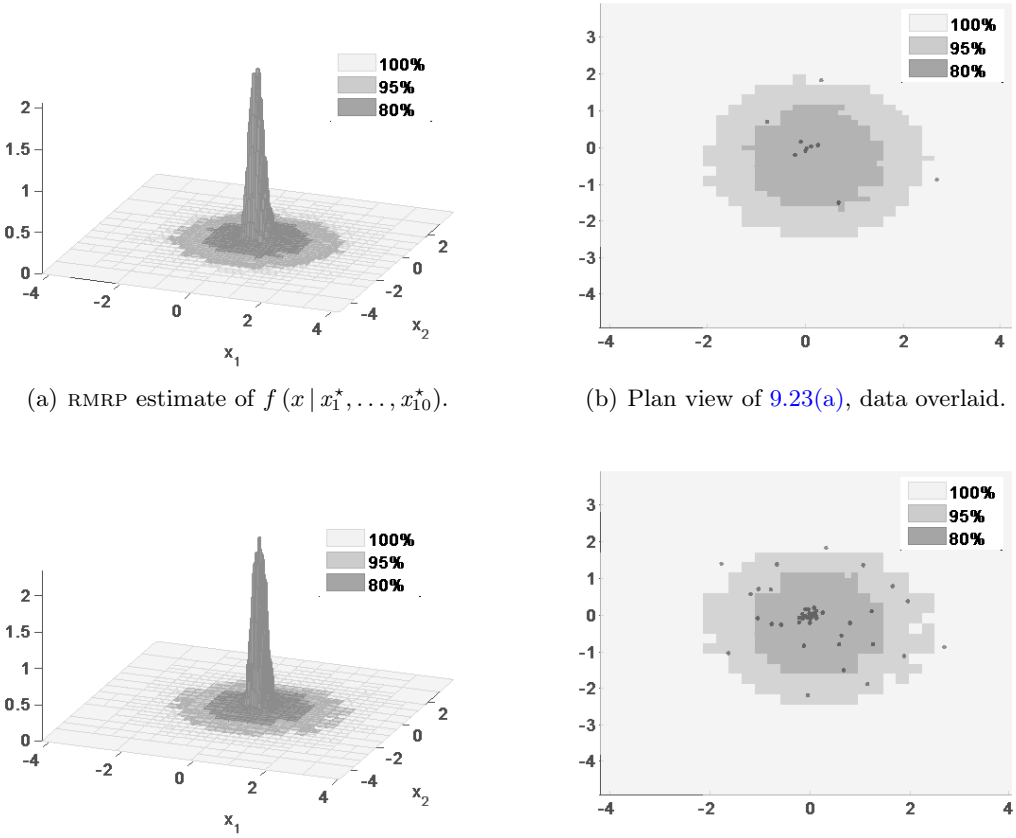
(a) RMRP estimate of $f(x \mid \mu^\star = (-1,1))$.

(b) RMRP estimate of $f(x \mid \mu^\star = (1,1))$.

Figure 9.22: Estimating $f(x \mid \mu^\star)$, closeups with observed values.

9.5.3.4 Estimating the posterior predictive density

Figure 9.23(a) shows the RPABC RMRP estimate of the posterior predictive density, with coverage regions, given the first 10 observations. This estimate was created using `SimulateData` and `Marginalise` as described in Section 9.3.5. The 10 observed data points are overlaid as points on the plan view of the coverage regions shown in Figure 9.23(b) (the plot has been simplified by omitting the lines indicating the partition of the RMRP, so that the data points can be seen more clearly). Figures 9.23(c) and 9.23(d) shows the equivalent plots for the RPABC estimate of the posterior predictive density given all 50 observations. Again these data points are overlaid on the plan view of the coverage regions of the density estimate.



(a) RMRP estimate of $f\left(x \mid x_1^\star, \ldots, x_{10}^\star\right)$.

(b) Plan view of 9.23(a), data overlaid.

(c) RMRP estimate of $f\left(x \mid x_1^\star, \ldots, x_{50}^\star\right)$.

(d) Plan view of 9.23(c), data overlaid.

Figure 9.23: Estimating the posterior predictive density.

9.5.3.5 RPABC timings

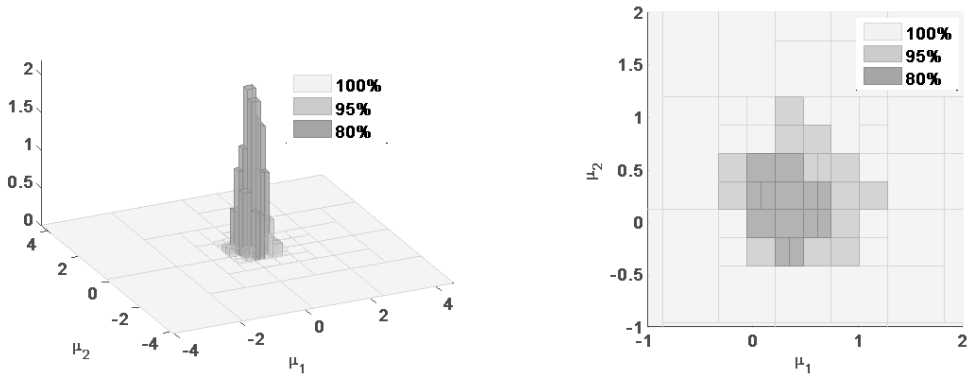
The initial step to check the prior support described in Section 9.3.1 took about 3 seconds. Generating $n = 100,000$ joint samples from the prior $f(\mu)$ and the model $f\left(x \mid \mu\right)$ took about a second. Forming the RMRP estimate of the joint density using the MCMC process took almost 7 hours. The joint density function is reasonably complex, as can be seen from Figure 9.17, and it took a long time for the chains used in the SRP MCMC process to converge.

The operations on the joint density estimate to obtain the posterior densities and the conditional density functions on various parameters of interest took about 15 seconds. Estimating each of the posterior predictive densities took about 12 hours. Although the final posterior predictive densities are only two-dimensional, the method used to compute them involves first creating another four-dimensional joint density estimate with values of μ drawn from the posterior, and then marginalising this (see Section 9.3.5). Almost all the time required for each posterior predictive estimate related to the SRP MCMC process for making the required joint density estimate.

9.5.4 Combining AABC and RPABC

This section shows the results of using the hybrid AABC-RPABC method described in Section 9.3.7.

An RMRP density estimate ${}^{\square}f^{\mu|\approx x^{\star}_{\cup 10}}$ of the approximate posterior $f\left(\mu \mid \approx x^{\star}_{\cup 10}\right)$ was created using a single iteration in the main AABC process and ε_1 such that 10% of 1,000,000 initial draws of μ' from the prior with $x' \sim f\left(x \mid \mu'\right)$ closest to $x^{\star}_{\cup 10}$ were retained. These samples, each together with its associated observation x', gave the simulation sample of size $n = 100,000$ from which to create an RMRP estimate of the posterior joint density. This was sliced on $x^{\star}_{\cup 10}$ and renormalised to give a refined posterior density estimate.



(a) Estimate of the posterior density after slicing.

(b) Plan view of Figure 9.24(a).

Figure 9.24: Highest posterior density regions after `MRPSlice`.

Figure 9.24(a) shows highest posterior density regions of the RMRP density estimate of $f\left(\mu \mid x^{\star}_{\cup 10}\right)$ after slicing ${}^{\square}f^{\mu|\approx x^{\star}_{\cup 10}}$ on $x^{\star}_{\cup 10}$. Figure 9.24(b) shows a plan view of the highest posterior density regions.

The 95% highest posterior density region is somewhat larger than that estimated using the full AABC method with $x^{\star}_{\cup 10}$ in Section 9.5.2. The limited partitioning in the AABC-RPABC posterior, compared with those in the AABC posterior, gives a relatively low resolution estimate. The hybrid RPABC-AABC method also took considerably longer. Obtaining the simulated samples took less than a minute, but forming the joint density estimate to be sliced to give the posterior density estimate took about 19 hours. This compares to a time of about an hour to obtain an estimate of $f\left(\mu \mid x^{\star}_{\cup 10}\right)$ using the full AABC method (see Section 9.5.2). Again, using more simulated samples in the density estimation process would improve the hybrid method results somewhat, but would also entail an even longer running time if the SRP MCMC method of Chapter 6 is used to form the RMRP joint density estimate.

9.6 Population genetics example in $\mathbb{R}^2$

This section demonstrates RPABC using simple example from population genetics. Results using AABC method of Beaumont et al. (2009) and the AABC-RPABC hybrid method are also shown. The example is artificial in the sense that the 'observed' data is simulated from a model using known parameter values.

9.6.1 The model and experimental setting

The aim was to make inferences about the coalescent-scaled mutation parameter θ (Wakeley 2009, chap. 4) for a single population using single nucleotide polymorphism (SNP) data (Wakeley 2009, chap. 1) from a group of present-day individuals.

The model $f(t \mid \theta)$ assumed a population evolving according to the Fisher-Wright neutral model (Wakeley 2009, chap. 3) and an infinite-sites (Kimura 1969) model of mutation with free recombination between the loci sampled, so that these loci were assumed to be unlinked (independent). Each locus modelled contained 25,000 nucleotide sites. The genealogical history (ancestral tree) of the individuals studied for each locus was randomly generated using an adaptation[1] of the `ms` program (Hudson 2002) assuming a population growth rate of 50% and no recombination or gene conversion within the locus (Hudson 2002). Mutations were simulated using `libsequence` (Thornton 2009). `libsequence` was also used to calculate summary statistics of the SNP data generated.

The number of present-day individuals used to simulate the artificial observed data was 10. The genetic sequence data simulated for a single locus for all 10 individuals gives a single realisation from the model: measures of the genetic diversity between the individual sequences may be used to make inferences about the scaled rate of mutation within the population. In this example only one measure of diversity, heterozygosity (Wakeley 2009, chap. 1), was used (i.e., $t \in \mathbb{R}$). Heterozygosity was calculated using `ThetaPi` in `libsequence` (Thornton 2009).

The 'observed' data was simulated from this model using $\theta = 0.2$ for $n_{obs} = 50$ loci. Under the model described this gave a sample of $n_{obs} = 50$ independent observations. The heterozygosity diversity measure was used to give the 1-dimensional summary statistic $t_i^\star$ for each observation $i = 1, \ldots, 50$. Summary statistics for the first 10 loci are listed in Section L.2 of Appendix L.

When the mutation rate is very low (and/or the growth rate very high) there will be very little genetic diversity and multiple simulations of loci data be summarised with exactly the same value of heterozygosity statistic. For this reason, a reasonably high value of θ was used for the true parameter value in this example. The prior support was also chosen to avoid the extreme 'spike' of heavily concentrated or repeated very small heterozygosity values that

[1] The adaptation, made with permission, was necessary in order to integrate `ms` with the model used for these simulations.

would be simulated if very low values of θ were included in the parameter space.

9.6.2 Inference using AABC

This section describes the results obtained using an implementation of the AABC method of Beaumont et al. (2009) summarised in Section 8.4.1. The prior density $f(\theta)$ was the Uniform(0.02, 2.00) density. The AABC process was used to obtain an estimate of the posterior density $f\left(\theta \mid t_i^\star\right)$ for each of the first 10 observations ($i = 1, \ldots, 10$) and for all these 10 loci summarised as a single observation using the average heterozygosity over the 10 loci, denoted by $t^\star_{\cup 10}$. The AABC process was also used to obtain an estimate of the posterior density for all 50 loci summarised as a single observation using the average heterozygosity over the 50 loci, denoted by $t^\star_{\cup 50}$.

The number of simulations used within each iteration was $n = 10,000$ and four iterations were carried out. The tolerance level ε_1 for the first iteration was set so that 100,000 values of θ' were simulated and 10% of these with associated t' closest to the summary statistic for the observed data $t^\star$ were retained. In subsequent iterations the tolerance levels were set as a fraction of the tolerance used in the previous iteration: $\varepsilon_2 = 0.75\varepsilon_1$, $\varepsilon_3 = 0.75\varepsilon_2$, $\varepsilon_4 = 0.50\varepsilon_3$.

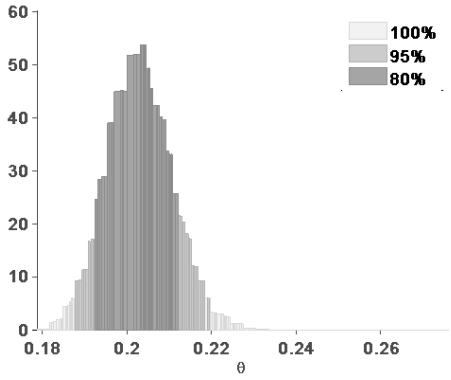
The AABC process is reasonably time-consuming, even for this simple example. The total number of simulations from the nucleotide model required over the four iterations of each individual AABC process was over 600,000 for each of the runs described in this section. Over an hour was required to obtain the simulations for the posterior given each of the individual $x_i^\star$. The time taken to simulate data from the model increased with the number of loci involved. Almost six hours was required for the simulations for the posterior given $t^\star_{\cup 10}$ and over 24 hours for the posterior given $t^\star_{\cup 50}$.

RMRP representations of the posterior densities were created using the MCMC process described in Chapter 6 with three chains. Each posterior density estimate was the average of 100 SRP samples collected, using a thin-out value of 1,000, from all three chains after convergence was diagnosed. It took about a minute to make each RMRP.

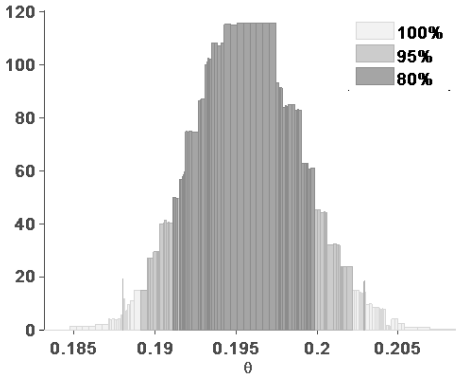
Figures 9.25(a) and 9.25(b) show the RMRP representations and highest posterior density regions for the AABC estimates of $f\left(\theta \mid t^\star_{\cup 10}\right)$ and $f\left(\theta \mid t^\star_{\cup 50}\right)$, respectively. The highest posterior density regions were found using the `CoverageRegion` operation (Algorithm 3.11). The 95% highest posterior density region from $f\left(\theta \mid t^\star_{\cup 10}\right)$ was [0.19, 0.22]. The 95% highest posterior density region from $f\left(\theta \mid t^\star_{\cup 50}\right)$ was [0.18, 0.21].

Figure 9.26 shows the RMRP estimates of the posteriors $f\left(\theta \mid t_1^\star\right)$ and $f\left(\theta \mid t_8^\star\right)$ obtained from AABC processes for the individual loci data $x_1^\star$ and $x_8^\star$, respectively. The same horizontal scale is used for both figures.

Figure 9.25 shows that, although the posterior density estimate is more concentrated when the summary statistic used in the AABC process is $t^\star_{\cup 50}$ (the average across 50 loci) than when

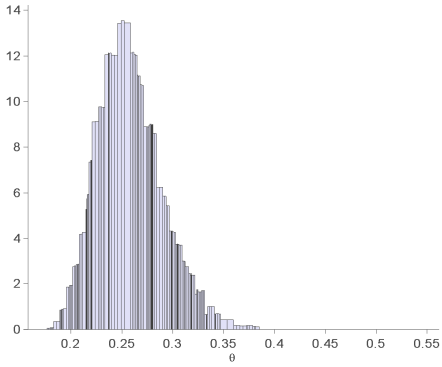


(a) Posterior density estimate given $t^{\star}_{\cup 10}$.

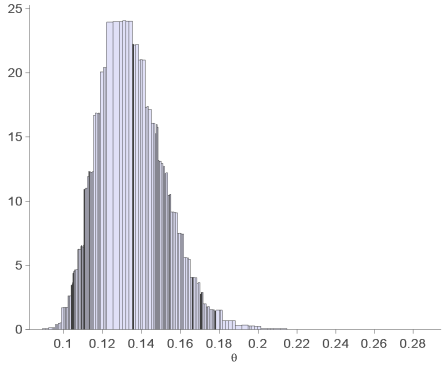


(b) Posterior density estimate given $t^{\star}_{\cup 50}$.

Figure 9.25: AABC posterior density estimates.



(a) Posterior density estimate given $t^{\star}_{1}$.



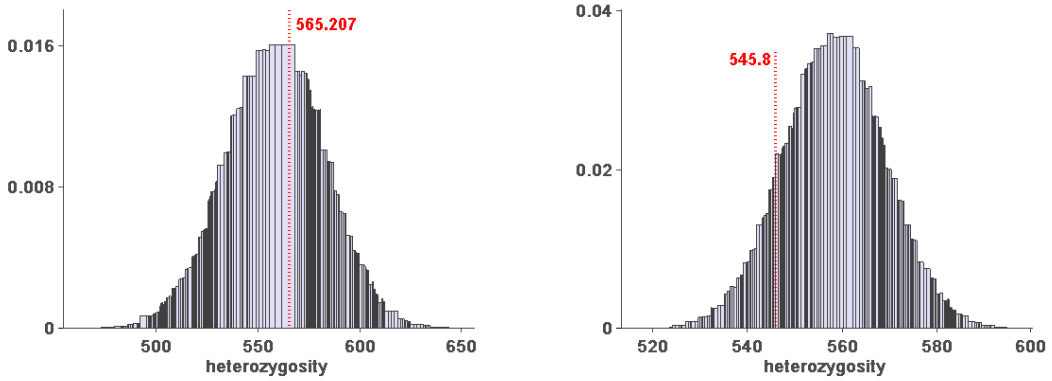
(b) Posterior density estimate given $t^{\star}_{8}$.

Figure 9.26: Posterior density estimates using individual loci.

the AABC summary statistic is $t^{\star}_{\cup 10}$ (the average across 10 loci), the mode of the posterior density estimate is somewhat below the value of $\theta = 0.2$ used to simulate the observed data.

The average summary statistic across the 50 loci constituting the observed data in this example is $t^{\star}_{\cup 50} = 545.800$. In this artificial example it is possible to use more simulations to examine how this value relates to the distribution of average heterozygosity statistic across 50 loci under the true parameter values. The model described in Section 9.6.1 with $\theta = 0.2$ was used to simulate 30,000 values of the average heterozygosity statistic and these simulations were used to construct an RMRP density estimate of $f\left(t_{\cup 50} \,|\, \theta^{\star} = 0.2\right)$. A similar process was also used to obtain an RMRP density estimate of $f\left(t_{\cup 10} \,|\, \theta^{\star} = 0.2\right)$.

Figure 9.27 shows these RMRP density estimates with the values of the statistics used in these AABC processes annotated. Figure 9.27(b) indicates that $t^{\star}_{\cup 50}$ is relatively small. The AABC process samples from the approximate posterior by trying to find values of θ that simulate data with summary statistics close to the observed value. This work well if the



(a) RMRP density estimate of $f\left(t_{\cup^{10}} \mid \theta^\star = 0.2\right)$. (b) RMRP density estimate of $f\left(t_{\cup^{50}} \mid \theta^\star = 0.2\right)$.

Figure 9.27: Density estimates for average summary statistics, $n = 30,000$.

observed value falls in the higher density regions of the sampling distribution of the statistic under the true model. In this case the observed value is sufficiently unusual to give the posterior density estimate shown in Figure 9.25(b).

9.6.3 Inference using RPABC

This section discusses the results obtained using an implementation of the RPABC process described in Section 9.3. The RPABC process was used to obtain an estimate of the posterior density $f\left(\theta \mid t_1^\star, \ldots, t_{10}^\star\right)$ (the posterior given summary statistics of the data for the first 10 loci as independent samples) and also for the posterior densities $f\left(\theta \mid t_1^\star, \ldots, t_{30}^\star\right)$ and $f\left(\theta \mid t_1^\star, \ldots, t_{50}^\star\right)$, to demonstrate the effect of more independent observations on the posterior.

The prior density $f(\theta)$ was the Uniform(0.020, 0.812) density. The prior support was calculated using the heuristic routine discussed in Section 9.3.1 and described in detail in Appendix K. The preliminary prior support used in the routine was [0.02, 2.00].

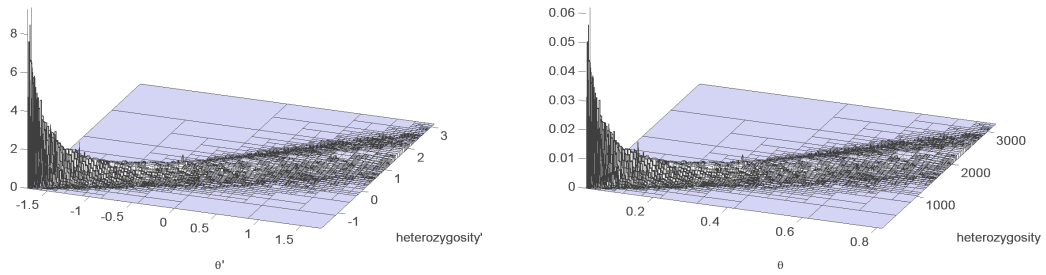
9.6.3.1 Estimating the joint density

The number of simulations used to estimate the joint density $f\left(\theta, t\right)$ was $n = 500,000$. The MCMC process used three chains. The joint density estimate was the average of 100 samples collected, using a thin-out value of 1,000, from all three chains after convergence was diagnosed.

Because of the very different scale of the parameter values and heterozygosity summary statistics, the RMRP estimate of the joint density was created using standardisations of both, as discussed in Section 4.4. The sample mean $\bar{\theta}'$ and sample standard deviation $s_{\theta'}$ of the sampled θ' were used to transform each θ' to $\frac{\theta' - \bar{\theta}'}{s_{\theta'}}$. Similarly the sample mean $\bar{t}'$ and sample standard deviation $s_{t'}$ of the associated t' were used to transform each t' to $\frac{t' - \bar{t}'}{s_{t'}}$. The RMRP operations, such as `MRPSlice`, were carried out on the RMRP estimate of the joint density in

the transformed coordinate space. The resulting RMRPs were back-transformed to give visualisations of the PCF density estimates in the original coordinate scales. The back-transformation is straightforward because the standardisation is a simple shift-and-scale and so the derivative of the inverse transformation used in the transformation formula (Wasserman 2003, chap. 2) is $\frac{1}{s_{\theta'}}$, i.e., just the constant by which the density values in the standardised coordinate space must be rescaled to obtain a normalised PCF in the original coordinate space.

Figure 9.28(a) shows the RMRP joint density estimate in the transformed coordinate space. Figure 9.28(b) shows the back-transformed PCF joint density estimate in the original coordinate space. All further illustrations in this section show back-transformed PCF density estimates in the original coordinate space unless noted otherwise.



(a) RMRP in the transformed coordinates.

(b) PCF in the original coordinates.

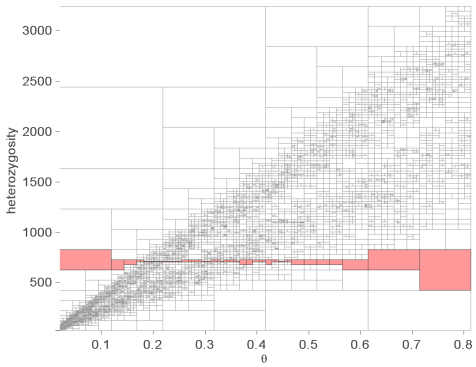
Figure 9.28: Estimating the joint density $f(\theta, t)$.

These visualisations of the joint density provide useful information about the relationship between θ and the heterozygosity genetic diversity statistic over a range of values of θ.

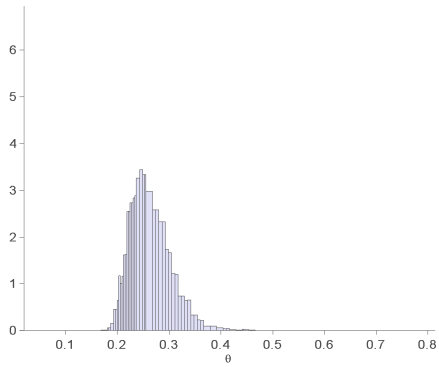
9.6.3.2 Estimating the posterior density

Figure 9.29 illustrates slicing to obtain the posterior, as described in Section 9.3.3, using $t_1^\star = 703.578$ and $t_8^\star = 368.622$, the largest and smallest, respectively, of the summary statistics for the first 10 loci. Figure 9.29(a) shows the elements of the partition of the support of the PCF joint density estimate through which the slice on $t_1^\star$ passes. Figure 9.29(b) shows the back-transformed PCF of the resulting RMRP estimate of the unnormalised conditional function $f(\theta \mid t_1^\star)$. The equivalent plots for $t_8^\star$ are show in Figures 9.29(c) and 9.29(d).

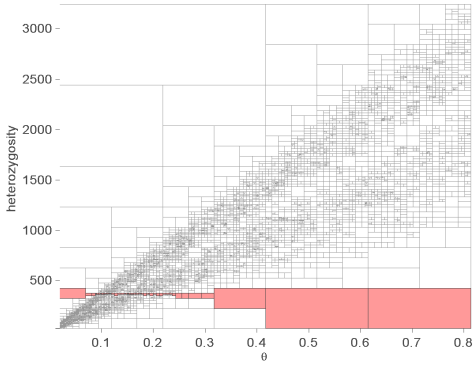
Figure 9.30 shows the RPABC PCF estimate of $f(\theta \mid t_1^\star, \ldots, t_{10}^\star)$, the posterior given the data for the first 10 loci, obtained by multiplying the slices on $t_1^\star, \ldots, t_{10}^\star$ and adjusting for $f(\theta)$ as described in Section 9.3.3. Figure 9.30(a) shows the posterior estimate relative to the support of the prior. Figure 9.30(b) shows the posterior estimate and highest posterior density regions on a rescaled horizontal axis so as to be more easily compared with Figure 9.25, the



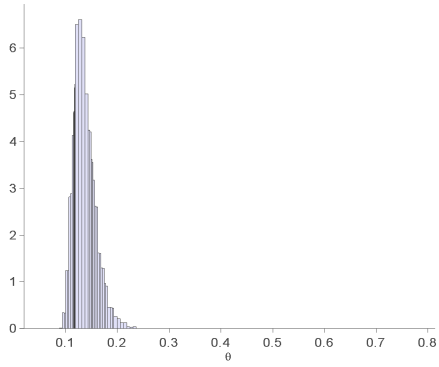
(a) Elements of the partition though which the slice on $t_1^\star$ passes.



(b) Unnormalised slice on $t_1^\star$.



(c) Elements of the partition though which the slice on $t_8^\star$ passes.



(d) Unnormalised slice on $t_8^\star$.

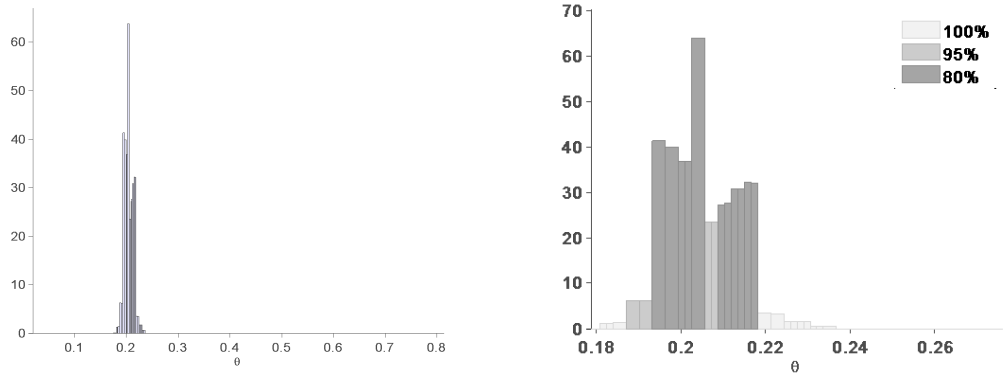
Figure 9.29: Slicing the joint density estimate.

PCF estimate of the equivalent posterior using AABC.

Figure 9.31 illustrates the effect of increasing the number of observations used to obtain the posterior density estimate, showing RPABC PCF estimates of the posterior density using the data for 30 and 50 loci on the same θ-scale as Figure 9.30(b).

The 95% highest posterior density region from the estimate of $f\left(\theta \mid t_1^\star, \ldots, t_{10}^\star\right)$ was [0.19, 0.22]. The 95% highest posterior density region from $f\left(\theta \mid t_1^\star, \ldots, t_{30}^\star\right)$ was [0.19, 0.21] and the 95% highest posterior density region from $f\left(\theta \mid t_1^\star, \ldots, t_{30}^\star\right)$ was [0.19, 0.21].

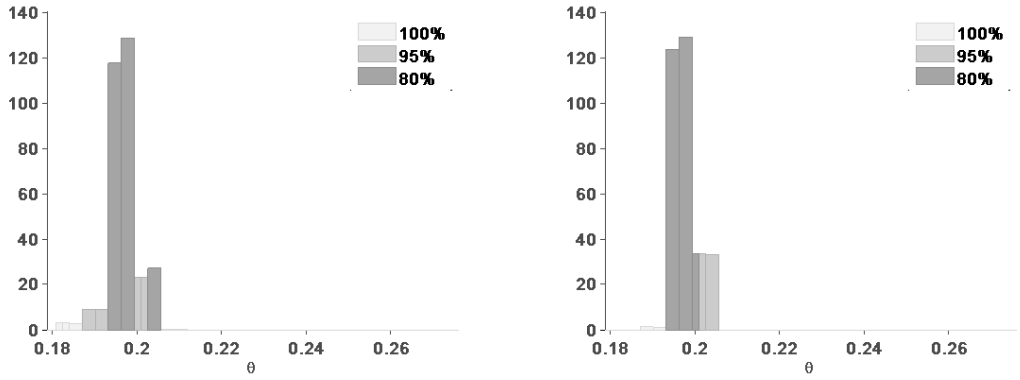
Exactly the same observed summary statistics for individual loci are used here as for the calculation of the averages on which the combined data AABC posteriors in Section 9.6.2 are based. However, if the observed data can be assumed to be n_{obs} independent realisations from the model, RPABC uses a *product likelihood* to estimate the posterior density $f\left(\theta \mid t_1^\star, \ldots, t_{n_{obs}}^\star\right)$ whereas AABC typically uses some form of combined data summary statistic to estimate $f\left(\theta \mid t_{\bigcup^{n_{obs}}}^\star\right)$. The product likelihood structure concentrates the posterior on



(a) On the scale of the prior support.

(b) Figure 9.30(a) rescaled.

Figure 9.30: Estimating $f\left(\theta \mid t_1^\star, \ldots, t_{10}^\star\right)$.



(a) Estimating $f\left(\theta \mid t_1^\star, \ldots, t_{30}^\star\right)$.

(b) Estimating $f\left(\theta \mid t_1^\star, \ldots, t_{50}^\star\right)$.

Figure 9.31: Posterior estimates with $n_{obs} = 30$ and $n_{obs} = 50$.

the values of θ having high likelihood given *all* the individual observations. Very unusual values in the sample will still affect the RPABC posterior, but it will also tend to concentrate if calculated using more independent observations whereas, as can be seen in Figure 9.25, the AABC posterior may shift, more than it concentrates, in response to more data. The RPABC posterior will not, of course, necessarily concentrate in the right location, just as the conventional product likelihood is not infallible.

9.6.3.3 What if? analysis

Figure 9.32(a) shows a PCF estimate of $f\left(t \mid \theta^\star = 0.2\right)$, the density of the summary statistics conditional on the true value of θ, obtained from the joint density estimate using `MRPSlice` and `Normalise` as described in Section 9.3.6. Figure 9.32(b) shows the same PCF on a restricted horizontal scale and with the values of the summary statistics for the 50 observed loci

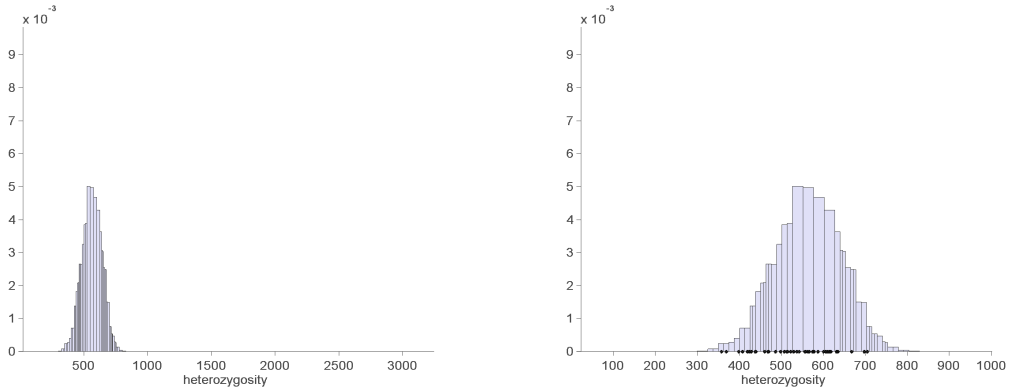
superimposed.

Figure 9.33(a) shows a PCF estimate of $f\left(t \,|\, \theta^\star = 0.12\right)$, the density of the summary statistics conditional on $\theta = 0.12$, again on the restricted scale and again with the values of the observed summary statistics superimposed. Similar estimates for $\theta^\star = 0.16$, $\theta^\star = 0.24$, and $\theta^\star = 0.28$ are shown in Figures 9.33(b), 9.33(c),and 9.33(d), respectively. Each of these conditional density estimates can be obtained very quickly from the single joint density created at the start of the RPABC process.

Table 9.2 shows the point log-likelihood estimates calculated using the product of the images, under each of the RMRP conditional density estimates, of the summary statistics for 10, 20, 30, 40, and 50 loci, as described in Section 9.3.6. For example, the estimate of the log-likelihood of $\theta^\star = 0.12$ using the data from the first 10 loci is

$$\hat{\ell}(0.12) = \sum_{i=1}^{10} \log\left({}^{\square}f^{t|0.12}(t_i^\star)\right)$$

where ${}^{\square}f^{t|0.12}$ is illustrated in Figure 9.33(a). The log-likelihood is higher for $\theta^\star = 0.2$, the true value of θ used to simulate the data from which these summary statistics were calculated, than for the other values of $\theta^\star$ used here for log-likelihoods calculated over the first 10, 20, 30, 40, and 50 loci (each row of Table 9.2).

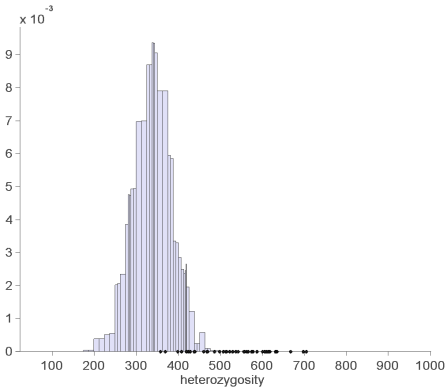


(a) On the scale of the joint density estimate

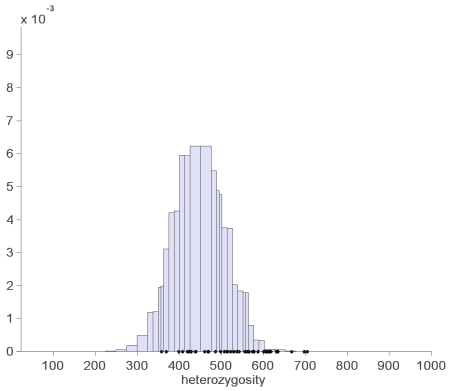
(b) Reduced horizontal scale.

Figure 9.32: PCF estimate of $f\left(t \,|\, \theta^\star = 0.2\right)$.

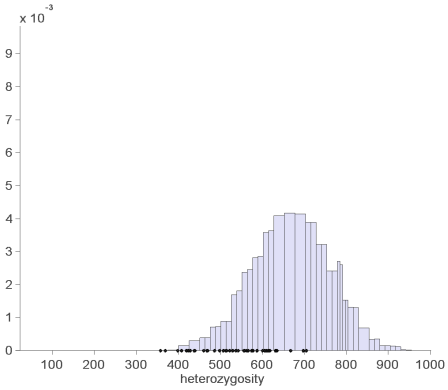
The density estimates created by slicing the joint density estimate (Figures 9.32 and 9.33) can be compared with the results of using direct simulation from the model to obtain similar 'what if' estimates. Figure 9.34 shows PCF density estimates and coverage regions created by simulating $n = 100,000$ summary statistics from the model $f\left(t \,|\, \theta^\star\right)$ for $\theta^\star = 0.2$ (the parameter value used to simulate the observed data summary statistics) and for $\theta^\star = 0.16$, $\theta^\star = 0.24$, and $\theta^\star = 0.28$. Each of the density estimates in Figure 9.34 took 5–10 minutes to create (including the simulation of the summary statistic values).



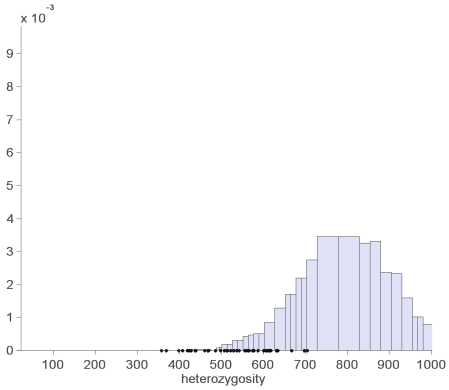
(a) PCF estimate of $f\left(t \,|\, \theta^\star = 0.12\right)$.



(b) PCF estimate of $f\left(t \,|\, \theta^\star = 0.16\right)$.



(c) PCF estimate of $f\left(t \,|\, \theta^\star = 0.24\right)$.



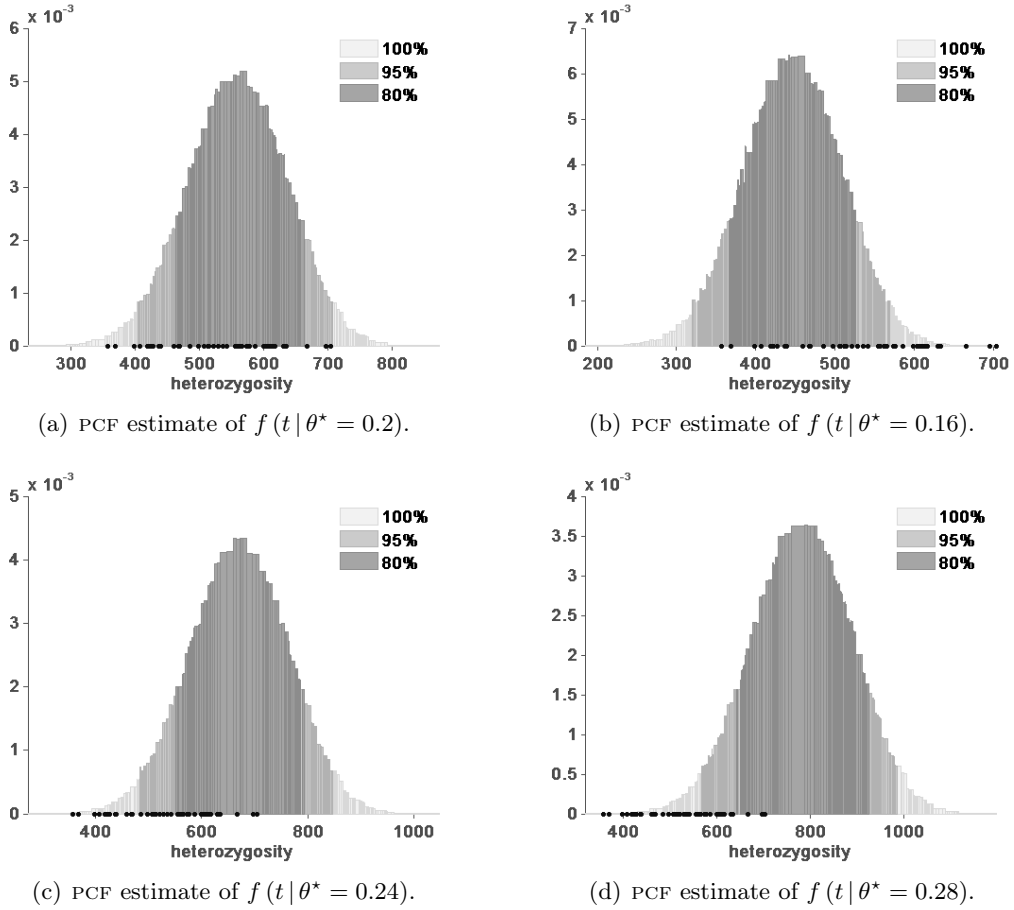
(d) PCF estimate of $f\left(t \,|\, \theta^\star = 0.28\right)$.

Figure 9.33: Estimating $f\left(t \,|\, \theta^\star\right)$ using slices of the joint density estimate.

Table 9.2: Point log-likelihoods for various values of $\theta^\star$.

$\theta^\star$	0.12	0.16	0.2	0.24	0.28	0.32	0.4
$\hat{\ell}_{10}(\theta^\star)$	−112.0	−21.9	6.2	−0.5	−17.5	−33.4	−56.9
$\hat{\ell}_{20}(\theta^\star)$	−169.3	−18.9	10.3	−10.8	−48.6	−86.4	−128.4
$\hat{\ell}_{30}(\theta^\star)$	−266.7	−40.1	15.3	−13.1	−67.8	−127.5	−189.2
$\hat{\ell}_{40}(\theta^\star)$	−367.8	−56.2	22.0	−11.4	−82.3	−161.0	−246.9
$\hat{\ell}_{50}(\theta^\star)$	−461.9	−62.8	29.9	−11.9	−110.3	−194.5	−305.0

The conditional density slices will be of coarser resolution than the estimates obtained by simulating directly from the model, but are very cheaply produced once the joint density estimate has been made. Especially if simulating from the model is more time-consuming, the best compromise might be to examine the RPABC posterior density estimate together with slices from the RPABC joint density for a large number of values of $\theta^\star$ to identify particular values of $\theta^\star$ to then use for more intensive simulations from the model as in Figure 9.34.



(a) PCF estimate of $f\left(t \,|\, \theta^\star = 0.2\right)$.

(b) PCF estimate of $f\left(t \,|\, \theta^\star = 0.16\right)$.

(c) PCF estimate of $f\left(t \,|\, \theta^\star = 0.24\right)$.

(d) PCF estimate of $f\left(t \,|\, \theta^\star = 0.28\right)$.

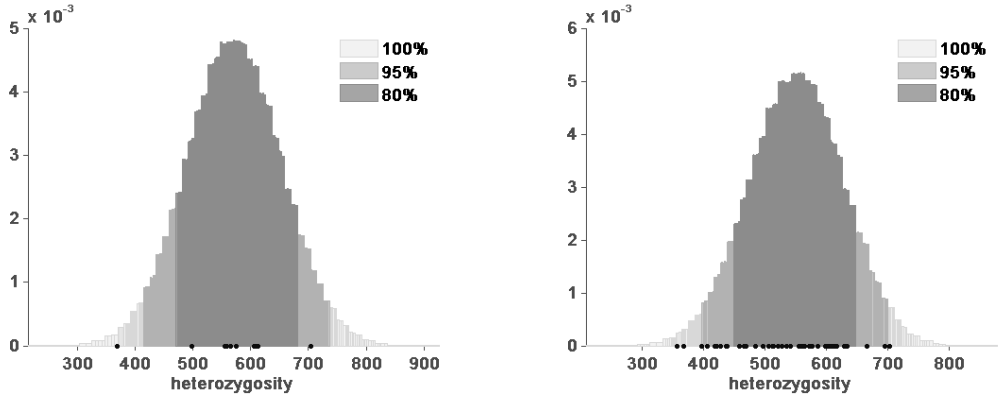
Figure 9.34: Estimating $f\left(t \,|\, \theta^\star\right)$ with 100,000 simulations directly from the model.

9.6.3.4 Estimating the posterior predictive density

Simulation from ${}^{\square}f^{\theta|t_1^\star,\ldots,t_{n_{obs}}^\star}$ using `SimulateData` (Algorithm 3.12) and `Marginalise` (Algorithm 3.10) can be used to create an estimate of the posterior predictive density $f\left(t \,|\, t_1^\star, \ldots, t_{n_{obs}}^\star\right)$, as described in Section 9.3.5.

Figure 9.35(a) shows the PCF estimate and coverage regions for the posterior predictive density given the summary statistics for the first 10 loci. These summary statistics are overlaid as points on the support of the density. Figure 9.35(b) shows the PCF estimate and coverage

regions of the posterior predictive density given the summary statistics for all 50 loci with the summary statistics overlaid.



(a) PCF estimate of $f\left(t \mid t_1^\star, \ldots, t_{10}^\star\right)$. (b) PCF estimate of $f\left(t \mid t_1^\star, \ldots, t_{50}^\star\right)$.

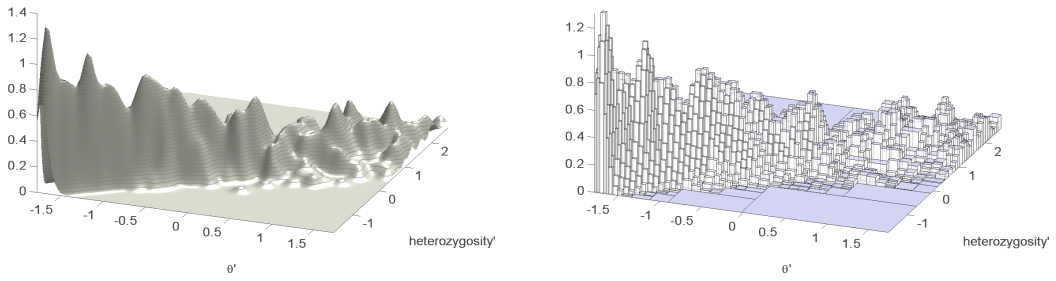
Figure 9.35: Estimating the posterior predictive density, with the observed summary statistics overlaid.

This example is not extended to include the use of simulated draws from the estimated posterior predictive density for Bayesian model checking. It would, however, be very straightforward to do this using the RMRP-structured posterior predictive density estimate. For example, $\texttt{SimulateData}\left({}^{\square}f^{t|t_1^\star,\ldots,t_{n_{obs}}^\star}\right)$ could be used to obtain a large number of samples of size n_{obs} from the posterior predictive density and statistics of these samples could then be compared with the equivalent statistics for the actual sample (O'Hagan and Forster 2004, chap. 8).

9.6.3.5 Approximating a KDE of the joint density

The KDE approximation was also tried as an alternative to the SRP MCMC method for forming the RMRP estimate of the joint density in the transformed coordinate space. In this example the simulation data available for forming the joint density estimate was quite challenging for any density estimation method because of the semi-discrete nature of the heterozygosity statistic and because of the extreme 'spike' in the density when θ is low, followed by the long spreading tail. The SRP MCMC method will tend to give an undersmoothed estimate with this kind of data, as Figure 9.28 demonstrates. The KDE produced by the multivariate MCMC-bandwidth kernel density estimator of Zhang et al. (2006) is probably oversmoothed.

Figure 9.36(a) shows a KDE of the joint density obtained using a sample size $n_K = 2,000$ and the multivariate MCMC-KDE method of Zhang et al. (2006) discussed in Chapter 7. The samples were standardised as described above before the KDE was formed. Figure 9.36(a) shows the KDE in this transformed coordinate space. Figure 9.36(b) shows an RMRP approximation to the KDE obtained using the method described in Section 7.3 and $\overline{\psi} = 0.0001$.

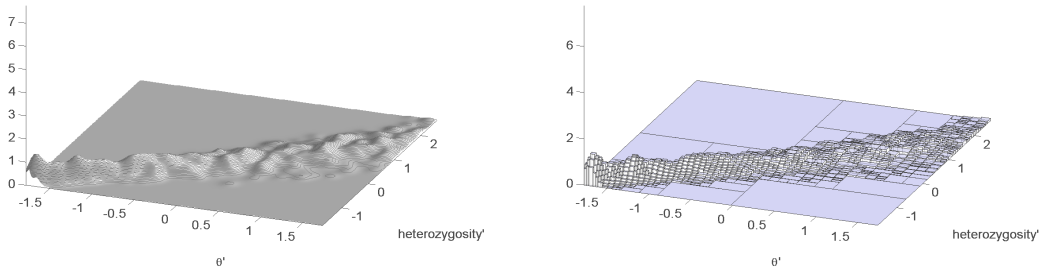


(a) KDE for $f(\theta, t)$. (b) RMRP approximation to the KDE.

Figure 9.36: Using a KDE to approximate the joint density (transformed coordinate space).

Figure 9.37(a) shows the KDE of Figure 9.36(a) with the same vertical scale as the RMRP estimate obtained using the MCMC SRP process shown in Figure 9.28(a). Figure 9.37(a) shows the RMRP approximation to the KDE similarly rescaled. The oversmoothing appears to be quite severe in the regions of the smaller values of θ that are most relevant to estimating the posterior in this example.

The analysis of the RMRP approximation of this KDE of the joint density is not continued further here but deserves much more thorough investigation. With a more suitable kernel density estimator the RMRP approximation to the KDE could be a much better joint density estimate on which to base the rest of the RPABC process than the SRP MCMC method is currently able to provide.



(a) KDE for $f(\theta, t)$. (b) RMRP approximation to the KDE.

Figure 9.37: Figure 9.36 rescaled.

9.6.3.6 RPABC timings

The initial step to determine a suitable prior support described in Section 9.3.1 took about 12 minutes. Generating $n = 500,000$ joint samples from the prior $f(\theta)$ and the model $f(t \mid \theta)$ took about 40-45 minutes. Forming the RMRP estimate of the joint density using the MCMC process took about 30 minutes. The entire process would take longer without the initial step to find the prior support (see Appendix K). The operations on the joint density estimate to obtain the posterior densities and the conditional density functions on various parameters of interest took less than 5 seconds. Estimating each of the posterior predictive densities took about 30 minutes.

9.6.4 Combining AABC and RPABC

This section describes the results obtained using the hybrid AABC-RPABC method described in Section 9.3.7. An RMRP density estimate ${}^{\square}f^{\theta|\approx t^{\star}_{\cup 10}}$ of the approximate posterior $f\left(\theta \mid \approx t^{\star}_{\cup 10}\right)$ was created using just one iteration in the main AABC process but with a larger number ($n = 50,000$) of simulated samples of θ' drawn from the approximate posterior than in Section 9.6.4. The number of initial draws from the prior was 500,000 and ε_1 was set so that 10% of these initial draws with $t' \sim f(t \mid \theta')$ closest to $t^{\star}_{\cup 10}$ and their associated summary statistics were retained for the sample of size $n = 50,000$ on which to base the AABC estimate of the posterior joint density $f\left((\theta,\, t) \mid \approx t^{\star}_{\cup 10}\right)$.

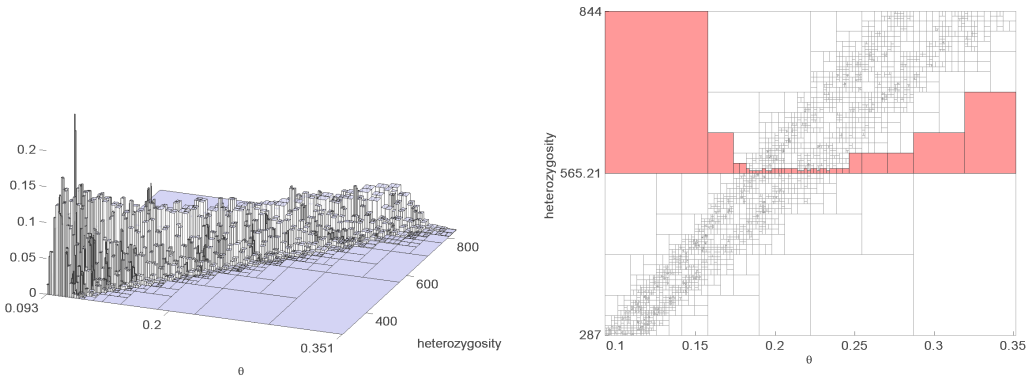
Figure 9.38(a) shows the PCF density estimate of this posterior joint density. Figure 9.38(b) shows, shaded, the elements of this PCF's partition through which the slice on $t^{\star}_{\cup 10}$ passes. Figure 9.39(a) shows the PCF density estimate of $f\left(\theta \mid \approx t^{\star}_{\cup 10}\right)$ obtained as a result of applying the AABC process with just one iteration. Figure 9.39(b) shows the refined PCF density estimate of $f\left(\theta \mid t^{\star}_{\cup 10}\right)$ after slicing the joint estimate shown in Figure 9.38(a) on $t^{\star}_{\cup 10} = 565.207$.

The RMRP estimate of the joint density was created using standardisations of the θ' and associated t' values, as in Section 9.6.3. The RMRP operations were carried out in the transformed coordinate space. The resulting RMRPs were back-transformed to give visualisations of the PCF density estimates on the original coordinate scales.

The estimate of $f\left(\theta \mid \approx t^{\star}_{\cup 10}\right)$ shown in Figure 9.39(a) is the result of just one iteration of the AABC process that gives a very loose approximation. The 95% highest posterior density region for the refined estimate of the posterior $f\left(\theta \mid t^{\star}_{\cup 10}\right)$ (the posterior given the combined data on 10 loci) shown in Figure 9.39(b) is [0.19, 0.23]. This is very similar to the 95% highest posterior density region of [0.19, 0.22] obtained from both the estimate using the full four AABC iterations and the first 10 loci shown in Figure 9.25(a) and from the RPABC RMRP estimate of $f(\theta \mid t^{\star}_1, \ldots, t^{\star}_{10})$ in Figure 9.30.

The time taken to obtain the estimate shown in Figure 9.39(b) was about 12 hours (almost all of which related to the time taken to simulate from the model in the initial, single, AABC iteration). This compares to a time of about 6 hours to obtain the AABC esti-

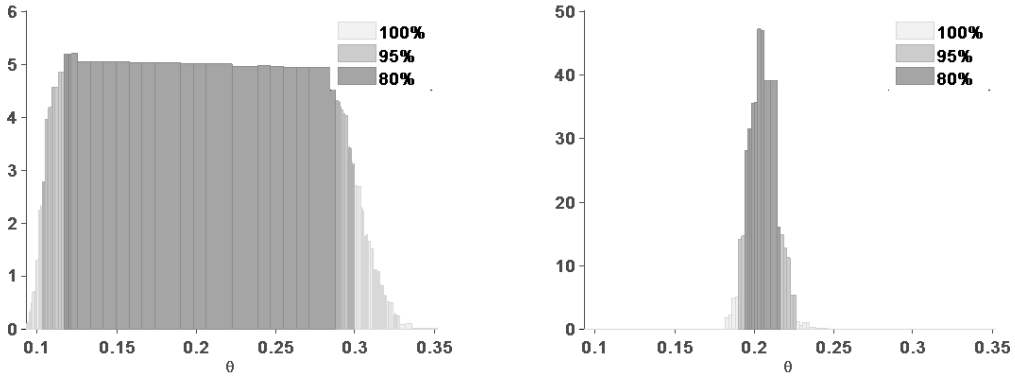
mate in Figure 9.25(a) and about 90 minutes to obtain RPABC estimates of $f(\theta \mid t_1^\star, \ldots, t_{10}^\star)$, $f(\theta \mid t_1^\star, \ldots, t_{30}^\star)$, and $f(\theta \mid t_1^\star, \ldots, t_{50}^\star)$ (Figures 9.30 and 9.31).



(a) The posterior joint density estimate.

(b) Path of the slice through the posterior joint partition.

Figure 9.38: Slicing the joint density estimate on $t_{\cup 10}^\star = 565.207$.



(a) Single AABC iteration estimate.

(b) Refined estimate after slicing.

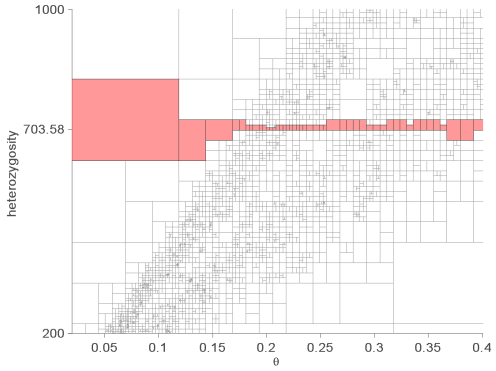
Figure 9.39: Refining the posterior density estimate using `MRPSlice`.

In this example, the combination of AABC and RPABC methods gives an equivalent result to that obtained from the full AABC procedure with multiple iterations but takes more time than either the AABC process for $x_{\cup 10}^\star$ or RPABC (discounting the time taken to obtain the RPABC posterior predictive density estimates).

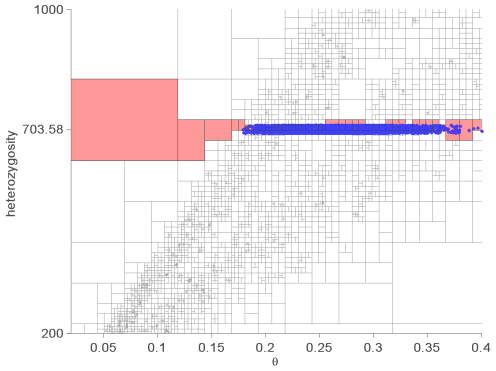
9.6.4.1 Relating AABC and RPABC

Figure 9.40 relates the RPABC process to the AABC process using as examples two of the first 10 loci, locus 1 and locus 8. Figure 9.40(a) shows a closeup of part of Figure 9.29(a), the path of the slice on the summary statistic for locus 1 through the RPABC joint density

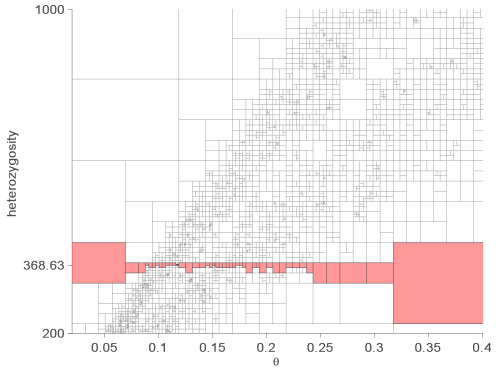
estimate (Figure 9.28(b)). Figure 9.40(b) shows the same closeup but superimposed with the 10,000 values of θ' simulated as draws from the AABC posterior $f(\theta \mid t_1^\star)$ for the AABC process described in Section 9.6.2. Figures 9.40(c) and 9.40(d) show similar plots using locus 8.



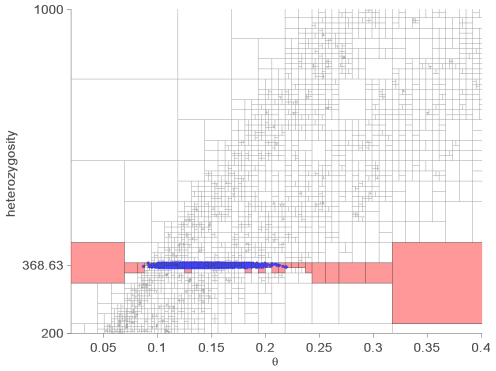
(a) Closeup of path for slice on $t_1^\star$.



(b) AABC samples overlaid on Figure 9.40(a).



(c) Closeup of path for slice on $t_8^\star$.



(d) AABC samples overlaid on Figure 9.40(c).

Figure 9.40: The relationship between posterior slices and AABC sampling.

For a single observed summary statistic $t^\star$, the sequence of tolerance values $\{\varepsilon_j\}$ used in the AABC process means that the AABC posterior is effectively created using a subset of the joint parameter-summary statistic space where the summary statistic values are uniformly close to $t^\star$. The AABC simulations (θ', t') are highly concentrated in this narrow band and are treated as though they are samples from $f(\theta \mid t^\star)$. By contrast, the RPABC slicing method bases the estimate of the posterior on a subset of the joint parameter-summary statistic space that has variable 'width', with respect to the coordinate(s) representing the summary statistic, depending on the density of the simulated (θ', t') in that region. Effectively this gives a narrow tolerance in the highest density regions, and a very wide tolerance in low density regions.

Although the number of simulated (θ', t') values used for the RPABC joint density estimate (500,000 in this case) is larger than the number of simulated values used to form the AABC posterior estimate (10,000), the RPABC simulations are spread over the entire joint parameter-

summary statistic space (as determined by the support of the prior used in the RPABC process) and the number of these simulations that actually fall in the whole of the shaded region in Figures 9.29(a) or 9.29(c) is likely to be very much less than 10,000. The much more refined partitions of the RMRP estimates for AABC posteriors formed using the SRP MCMC method reflect this. Creating the RPABC RMRP joint density estimate by approximating a KDE very tightly could give much smoother RMRP estimates for the RPABC posteriors.

9.7 Population genetics example in $\mathbb{R}^4$

This section explores a higher-dimensional example from population genetics. The example is again artificial in the sense that the 'observed' data is simulated from a model using known parameter values.

9.7.1 The model and experimental setting

The aim was to make inferences about both the coalescent-scaled mutation parameter θ and the percentage population growth rate g (Wakeley 2009, chap. 4) for a single population using SNP data from a group of present-day individuals. To avoid confusion between θ as the mutation parameter in this model and θ as a general symbol for a vector of parameters of interest, both elements in 2-dimensional vector of parameters of interest, (θ, g), are explicitly given in the formulas and expressions used in this section.

The model $f(t \mid (\theta, g))$ was identical to that used in Section 9.6 except that the population growth parameter used in the simulation of genealogical histories (ancestral trees) was one of the parameters of interest in the inference process. Mutations were again simulated using `libsequence` (Thornton 2009). `libsequence` was also again used to calculate summary statistics of the SNP data generated.

The number of present-day individuals used to simulate the artificial observed data was 10. The genetic sequence data simulated for a single locus for all 10 individuals gives a single realisation from the model. Two measures of diversity were used: heterozygosity, and the number of segregating sites in the sample (Wakeley 2009, chap. 1). The 2-dimensional vector of summary statistics is referred to as t.

The 'observed' data was simulated from this model using $\theta = 0.2$ and $g = 50$ for $n_{obs} = 50$ loci. Under the model described this gave a sample of $n_{obs} = 50$ independent observations. The heterozygosity and segregating sites diversity measures were used to give the 2-dimensional summary statistic $t_i^\star$ for each observation $i = 1, \ldots, 50$. The summary statistics of the first 10 loci are listed in Section L.2 of Appendix L.

The segregating sites summary statistic for an individual locus is a discrete variable. When the mutation rate θ is reasonably high and the population growth rate is low, the range of the segregating sites statistic is wide, but when very large numbers of samples of this variable

are required there will be many repeated realisations of the same discrete values and the samples will be unsuitable for use with density estimation methods intended for continuous, not discrete, data. The *average* segregating site statistic over a reasonable number of different loci can, however, be treated as a continuous variable.

9.7.2 Inference using AABC

This section describes the results obtained using an implementation of the AABC method of Beaumont et al. (2009) summarised in Section 8.4.1. The prior density for the mutation rate parameter, $f(\theta)$, was the Uniform(0.02, 0.20) density. The prior density for the population growth rate parameter, $f(g)$, was the Uniform(30, 70) density.

The AABC process was used to obtain an estimate of the posterior density $f\left((\theta, g) \,|\, t^{\star}_{\cup 10}\right)$, the posterior given the average summary statistic over the first 10 loci. An AABC process using the average statistics over all 50 loci was attempted but the time taken to generate the required number of samples of θ was prohibitive.

The number of simulations used within each iteration was $n = 10,000$ and four iterations were carried out. The tolerance level ε_1 for the first iteration was set so that 100,000 values of $(\theta', \boldsymbol{g}')$ were simulated and 10% of these with associated t' closest to the summary statistic for the observed data $t^{\star}$ were retained. In subsequent iterations the tolerance levels were set as a fraction of the tolerance used in the previous iteration: $\varepsilon_2 = 0.75\varepsilon_1$, $\varepsilon_3 = 0.75\varepsilon_2$, $\varepsilon_4 = 0.50\varepsilon_3$ (exactly as in Section 9.6.2).

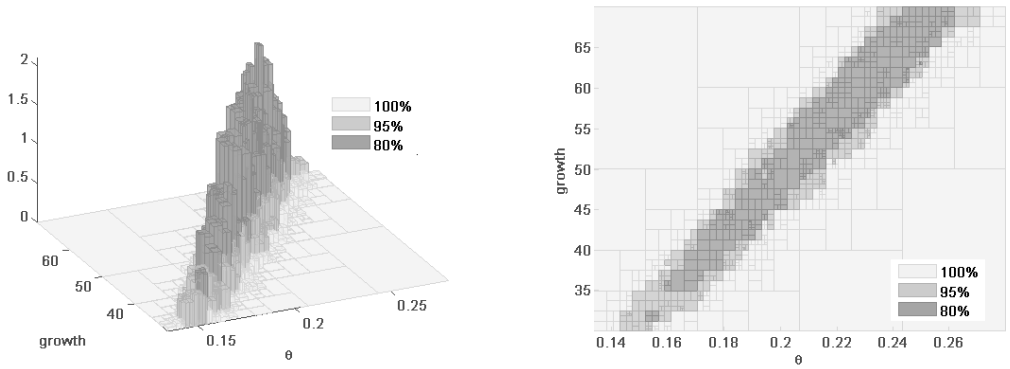
In total, more than two million simulations from the nucleotide model were required over the four iterations of the AABC process. Generating the required samples took about 16 hours.

An RMRP representation of the posterior density was created using the MCMC process described in Chapter 6 using three chains. The posterior density estimate was the average of 100 samples collected, using a thin-out value of 1,000, from all three chains after convergence was diagnosed. The difference in scale between θ and g required that the values of the θ and g sampled using the AABC process be standardised. A similar standardisation process to that described in Section 9.6.3 in relation to the values of heterozygosity and θ was used. All RMRP operations were carried out on the RMRPs in this transformed space. The PCF density estimates displayed in this section were created by back-transforming the RMRPs into the original coordinate space.

Figure 9.41(a) shows the PCF representation and highest posterior density regions of the AABC estimate of $f\left((\theta, g) \,|\, t^{\star}_{\cup 10}\right)$. Figure 9.41(b) shows the highest posterior density regions in plan view. The range of values for g is clearly restricted by the support of the prior for g.

9.7.3 Inference using RPABC

It was not possible apply the full RPABC method described in Section 9.3 in this setting using the available nucleotide model and the summary statistics of the nucleotide data incorporated



(a) Posterior density and highest posterior density regions.

(b) Plan view of Figure 9.41(a).

Figure 9.41: AABC posterior density estimate of $f\left((\theta, g) \,|\, x^{\star}_{\cup 10}\right)$.

into that model. Section 9.7.1 notes that the per-locus segregating sites summary statistic is a discrete variable. This naturally caused severe problems for the SRP MCMC process used to make the density estimate of the joint parameter-summary statistic samples. The SRP MCMC method is intended for use with continuous data. The alternative of forming the RMRP estimate of the joint density by approximating a KDE was also precluded because KDEs are also only suitable for continuous data. It would have been possible, but disproportionately time-consuming relative to the importance of this example in the context of this thesis as a whole, to adapt the model to use more suitable summary statistics.

The AABC method was not adversely affected by the discrete nature of the segregating sites summary statistic. This is partly because the average value of this statistic over 10 loci can be treated as a continuous variable (the range of values of θ and growth considered in this example avoid the point-mass concentrations of very small segregating sites values that indicate very low genetic diversity). In addition, AABC does not require an estimate of the joint density: the only density estimates involved are estimates of the posterior itself.

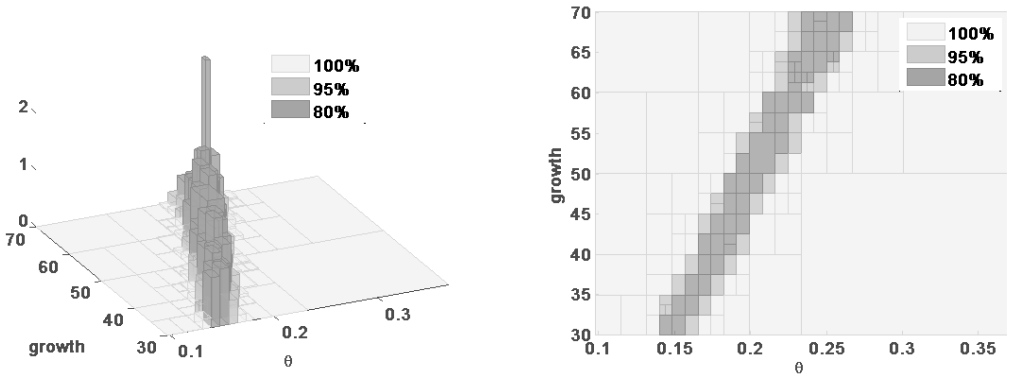
9.7.4 Combining AABC and RPABC

The segregating sites summary statistic also caused fewer problems for the hybrid AABC-RPABC method described in Section 9.3.7 when this was applied using an initial AABC iteration approximating $f\left((\theta, g) \,|\, \approx x^{\star}_{\cup 10}\right)$ because the summary statistics then used in the joint density estimate are averages over 10 loci.

A single iteration of the AABC process was used with ε_1 such that 10% of 1,000,000 initial draws of $(\theta', \boldsymbol{g}')$ from the priors with $t' \sim f\left(t \,|\, (\theta', \boldsymbol{g}'\right)$ closest to $t^{\star}_{\cup 10}$ were retained. The retained values of θ', $\boldsymbol{g}'$, each with the associated summary statistic t', gave a sample of 100, 000 joint tuples $(\theta', \boldsymbol{g}', t')$ from the approximate posterior joint density $f\left((\theta, g, t) \,|\, \approx x^{\star}_{\cup 10}\right)$. An

RMRP density estimate ${}^{\square}f^{(\theta,g,t)|\approx x^{\star}_{\cup 10}}$ of the approximate posterior joint density was created and `MRPSlice` and `Normalise` were then used to obtain the AABC-RPABC estimate of the posterior $f\left((\theta, g) \mid t^{\star}_{\cup 10}\right)$.

Figure 9.42(a) shows highest posterior density regions of the RMRP density estimate of $f\left((\theta, g) \mid t^{\star}_{\cup 10}\right)$ after slicing ${}^{\square}f^{(\theta,g,t)|\approx x^{\star}_{\cup 10}}$ on $t^{\star}_{\cup 10} = (2048.5, 565.207)$. Figure 9.42(b) shows a plan view of the highest posterior density regions. The 95% highest posterior density region of the AABC-RPABC estimate of the posterior density was similar to that obtained using the full AABC method in Section 9.7.2. Again, however, the partitioning of the root box of the final PCF density estimate was not as fine as that given by the full AABC method.



(a) Estimate of the posterior density after slicing.

(b) Plan view of Figure 9.42(a).

Figure 9.42: Highest posterior density regions after `MRPSlice`.

Obtaining the simulated samples using the single AABC iteration took about 10 hours. One million simulations from the model were required, compared with over two million for the full AABC method (see Section 9.7.2), but the average time taken per simulation is higher when a wider range of values of θ and g is used. The number of joint samples used ($n = 100,000$) is too small for the four-dimensional joint parameter-summary statistic space but even this sample size meant that forming the joint density estimate to be sliced to give the posterior density estimate took a further 10 hours using the SRP MCMC method. This compares to a total time of about 16 hours to obtain an estimate of $f\left((\theta, g) \mid x^{\star}_{\cup 10}\right)$ using the full AABC method. Simulating a larger sample size n for the joint density estimate might improve the hybrid method results somewhat, but would add to both the data-sampling time and the running time of the SRP MCMC method.

The alternative of obtaining the RMRP joint density estimate by approximating a KDE was explored, but again the multivariate MCMC bandwidth KDE of Zhang et al. (2006) was unsuited to the complexity of the joint distribution. The oversmoothing in the KDE produced with this four-dimensional data was considerably worse than that in $d = 2$ (Section 9.6.3). However, the KDE approximation method could still provide the best way to form the RMRP joint density

estimate required by the RPABC and AABC-RPABC methods, if a suitable multivariate kernel density estimator is used.

9.8 Discussion

The examples discussed in this chapter provide a 'proof of concept' of the RPABC method for Bayesian inference with complex models and intractable likelihoods described in Section 9.3. It has been shown that the RMRP operations `MRPSlice` and `Marginalise` can be combined with RMRP arithmetic to obtain an RMRP posterior density estimate that exploits the product likelihood structure, and that the time taken to do this is minimal once the RMRP estimate of the joint parameter-data or parameter-summary statistic density has been formed. Visualisations of the RMRP joint density estimate can be easily obtained even for high dimensional problems and provide useful additional information about the relationships between the parameters and data. Density estimates conditional on selected parameter values sliced out of the joint density estimate can also be used to get rapid point-likelihood approximations. Simulating from the posterior density estimate using the RMRP operation `SimulateData` is also straightforward can be used to obtain an estimate of the posterior predictive density.

The main limitation of the RPABC method as it is currently implemented and demonstrated in this chapter is the reliance on the SRP MCMC process as the means of obtaining the estimate of the joint parameter-data or parameter-summary statistic density. As is discussed in Chapter 6, this method begins to struggle severely in four or five dimensions, requiring very large data sets to achieve reasonable estimation errors in these dimensions but also taking an increasingly long time to achieve convergence as the size of the data set and the number of dimensions grows. The results of the four dimensional examples in this chapter (Section 9.5 and 9.7) were very much affected by this limitation. In both cases it would have been interesting to have been able to use much more simulated sample data to obtain the joint density estimate and to investigate whether this improved the final posterior density estimates, but these much larger data sets proved impractical in conjunction with the SRP MCMC process.

The KDE approximation method described in Chapter 7 may offer the best means of obtaining the RMRP joint density estimate required by RPABC. Although further work on the SRP MCMC process may mitigate some of the current issues, approximating a smooth KDE very closely will give a smoother, much more finely partitioned, RMRP than is likely to be easily obtained using the SRP MCMC process. A KDE approximation would also avoid a reliance on very large data sets. Simulation from the kind of very complex models that may require these forms of inference technique can be very time-consuming, as even the relatively simple examples in Sections 9.4 and 9.5 demonstrate. Multivariate kernel density estimation in four or five dimensions, and above, is itself very challenging, but — as has already been concluded in Chapter 7 — approximating a kernel density estimate made using some of the newer fast

multivariate density estimation techniques being developed in fields such as data mining has considerable potential to provide a much improved high dimensional RMRP density estimate.

RPABC is much more affected by dimensionality than AABC. RPABC requires a density estimate in the full joint parameter-summary statistic space whereas AABC only uses density estimation to visualise and analyse the posterior density estimate on the parameter space. In addition, the AABC method naturally concentrates a large number of simulation samples from the posterior onto a small support. The purpose of the artificial examples in this chapter has been to illustrate the main differences, and similarities between RPABC and AABC rather than to compare their relative effectiveness. Such a comparison can only be carried out when the limitations imposed on RPABC by the method of joint density estimation have been satisfactorily solved.

The hybrid AABC-RPABC method discussed in this chapter may be of some interest as a way of creating a posterior density estimate from an AABC approximate posterior, but the examples show that this approach is also limited by the difficulty of creating the joint density estimate. In low dimensional examples the method performed well, but the full RPABC method was also very efficient and gave a wider range of inference opportunities. In higher dimensions the hybrid inherited the weaknesses, rather than the strengths, of its parents.

Despite the issues summarised in this section the final conclusion of this chapter is that the RPABC method has considerable potential to provide a powerful and innovative new approach to the challenging problem of simulation-intensive inference with complex models and intractable likelihoods. The ability of RPABC to exploit the product likelihood structure when independent observations are available and the rich range of inference methods that can be applied using the RPABC RMRP operations are both strong advantages and justify further research and development of this method.

Chapter 10

Summary and conclusions

The overall purpose of this thesis has been to explore the potential for density estimation using regular pavings. Chapter 3 has formalised and explained the wide range of operations that may be carried out on a piecewise constant density estimate structured as a real-mapped regular paving (RMRP).

Chapter 4 discusses the implications of the restrictions that apply to RMRP density estimates. Other research has already shown that, despite these restrictions, the SRP extension of an RP can be used in data-adaptive partitioning algorithms. Chapter 6 investigates one such algorithm, the SRP MCMC of Sainudiin et al. (2013), in detail. The semi-automatic method of assessing convergence of the Markov chains proposed in Chapter 6 addresses one of the weaknesses of this method of forming an RMRP density estimate. The method is shown to work well with low-dimensional data but to struggle with data in four or five dimensions. Although many other density estimation methods experience similar problems, this limits the applications for which RMRP density estimates are currently suitable.

The KDE approximation algorithm described in Chapter 7 has been developed in response to the limitations of the SRP MCMC partitioning method. Approximating a KDE using an RMRP offers the potential to be able to combine the superior convergence properties of a KDE with the operational properties of an RMRP. The algorithm is shown to be able to approximate a KDE reasonably closely. However, testing of the algorithm has been limited to approximating KDEs produced by a single kernel density estimator, which is itself best suited to lower dimensional data. The next step required to progress this method further is to carry out more extensive testing with a multivariate kernel density estimator able to perform well with complex densities in higher dimensions. Finding such an estimator appears to be the chief challenge that will have to be overcome in order to expand the scope for density estimation using RMRPs.

Chapter 9 proposes and demonstrates a new method for simulation-intensive Bayesian inference for complex models with intractable likelihoods that exploits the arithmetical and other operational properties of RMRP-structured density estimates. Unlike the adaptive approximate Bayesian computation approach that is increasingly popular for tackling these types of inference problems, the regular paving approximate Bayesian computation method proposed in Chapter 9 is able to approximate a product-likelihood posterior density estimate. However, the method is in an early stage of development and is currently limited by the unsuitability of the SRP MCMC process to higher dimensional data. Chapter 9 provides the foundation for further work on regular paving approximate Bayesian computation.

Abbreviations

Acronyms

AABC adaptive approximate Bayesian computation.

ABC approximate Bayesian computation.

CFTP coupling from the past.

IAE integrated absolute error.

ISE integrated squared error.

KDE kernel density estimate.

MCMC Markov chain Monte Carlo.

MIAE mean integrated absolute error.

MISE mean integrated squared error.

MRP mapped regular paving.

PCF piecewise-constant function.

PSRF potential scale reduction factor.

RMRP real-mapped regular paving.

RP regular paving.

RPABC regular paving approximate Bayesian computation.

RPQ randomised priority queue.

SEB statistically equivalent block.

SRP statistical regular paving.

Appendix A

Technical specifications

A.1 MRS: a C++ class library for statistical set processing

The structures and algorithms developed as part of this thesis are implemented as part of MRS*: a C++ class library for statistical set processing* and publicly available under the terms of the GNU General Public License from http://www.math.canterbury.ac.nz/~r.sainudiin/codes/mrs/.

A.2 Computer hardware and operating system

Unless otherwise stated all results described in this thesis were obtained on a system with the following technical specifications:

Model: Supermicro 6016GT compute server.

CPU: 2 x Intel Xeon X5670 [Six-Core 2.93 GHz] CPUs.

Kernel: Linux.

Operating system: CENSUSES version 11.2 (x86-64).

A.3 Other computer software used

Unless otherwise stated all figures in this thesis were created using MATLAB® R2011b.

Appendix B

Example densities

B.1 Density I

Density I is a mixture of two multivariate Normal densities for $x \in \mathbb{R}^d$. Density I has no correlation between data coordinates and moderate bimodality:

$$f_I(x \,|\, \mu_a, \Sigma_a, \mu_b, \Sigma_b) = \frac{1}{2}\varphi(x \,|\, \mu_a, \Sigma_a) + \frac{1}{2}\varphi(x \,|\, \mu_b, \Sigma_b),$$

where $\varphi(x \,|\, \mu, \Sigma)$ is the multivariate Normal density with mean $\mu \in \mathbb{R}^d$ and $d \times d$ variance-covariance matrix Σ, and

$$\mu_a = \begin{pmatrix} 1.0 \\ 0.0 \\ \vdots \\ 0.0 \end{pmatrix}, \quad \Sigma_a = \begin{pmatrix} \sigma_a(x_1, x_1) & 0 & \cdots & 0 \\ 0 & \sigma_a(x_2, x_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_a(x_d, x_d) \end{pmatrix},$$

$$\mu_b = \begin{pmatrix} 2.5 \\ \vdots \\ 2.5 \end{pmatrix}, \quad \Sigma_b = \begin{pmatrix} \sigma_b(x_1, x_1) & 0 & \cdots & 0 \\ 0 & \sigma_b(x_2, x_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_b(x_d, x_d) \end{pmatrix},$$

and

$$\sigma_a(x_i, x_i) = \frac{1.5}{1 + \left(\frac{i-1}{2}\right)}, \sigma_b(x_i, x_i) = \frac{0.625}{1 + \left(\frac{i-1}{4}\right)} \; i = 1, \ldots, d, .$$

When $d = 2$ Density I is a mixture of two bivariate Normal densities with

$$\mu_a = \begin{pmatrix} 1.0 \\ 0.0 \end{pmatrix}, \quad \Sigma_a = \begin{pmatrix} 1.5 & 0 \\ 0 & 1.0 \end{pmatrix}, \quad \mu_b = \begin{pmatrix} 2.5 \\ 2.5 \end{pmatrix}, \quad \Sigma_b = \begin{pmatrix} 0.625 & 0 \\ 0 & 0.5 \end{pmatrix}.$$

B.2 Density II

Density II is a mixture of two multivariate Normal densities for $x \in \mathbb{R}^d$. Density II has high correlation between data coordinates and high bimodality:

$$f_{II}(x \,|\, \mu_a, \Sigma_a, \mu_b, \Sigma_b) = \frac{1}{2}\varphi(x \,|\, \mu_a, \Sigma_a) + \frac{1}{2}\varphi(x \,|\, \mu_b, \Sigma_b),$$

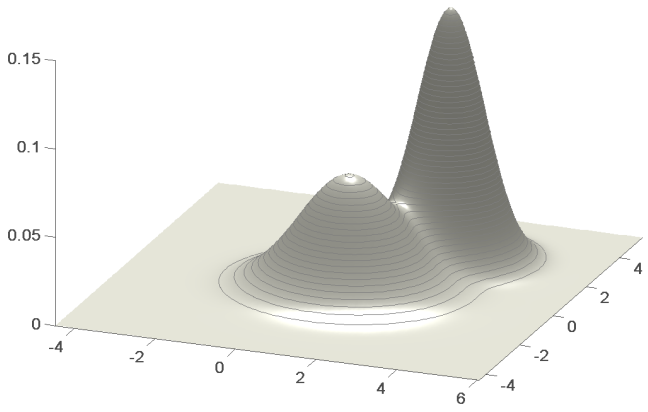


Figure B.1: Density I, $d = 2$.

where $\varphi(x \,|\, \mu, \Sigma)$ is the multivariate Normal density with mean $\mu \in \mathbb{R}^d$ and $d \times d$ variance-covariance matrix Σ, and

$$\mu_a = \begin{pmatrix} 2.0 \\ \vdots \\ 2.0 \end{pmatrix}, \quad \Sigma_a = \begin{pmatrix} \sigma_a(x_1, x_1) & \sigma_a(x_1, x_2) & \cdots & \sigma_a(x_1, x_d) \\ \sigma_a(x_2, x_1) & \sigma_a(x_2, x_2) & \cdots & \sigma_a(x_2, x_d) \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_a(x_d, x_1) & \sigma_a(x_d, x_2) & \cdots & \sigma_a(x_d, x_d) \end{pmatrix},$$

$$\mu_b = \begin{pmatrix} -1.5 \\ \vdots \\ -1.5 \end{pmatrix}, \quad \Sigma_b = \begin{pmatrix} \sigma_b(x_1, x_1) & \sigma_b(x_1, x_2) & \cdots & \sigma_b(x_1, x_d) \\ \sigma_b(x_2, x_1) & \sigma_b(x_2, x_2) & \cdots & \sigma_b(x_2, x_d) \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_b(x_d, x_1) & \sigma_b(x_d, x_2) & \cdots & \sigma_b(x_d, x_d) \end{pmatrix},$$

and

$$\sigma_a(x_i, x_j) = \begin{cases} 1 & \text{if } i = j, \\ -0.9^{|i-j|} & \text{if } i \neq j, \end{cases}, \quad \sigma_b(x_i, x_j) = \begin{cases} 1 & \text{if } i = j, \\ 0.3^{|i-j|} & \text{if } i \neq j, \end{cases}.$$

When $d = 2$ Density II is a mixture of two bivariate Normal densities and is the same as Density A studied in Zhang et al. (2006):

$$\mu_a = \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \quad \Sigma_a = \begin{pmatrix} 1 & -0.9 \\ -0.9 & 1 \end{pmatrix}, \quad \mu_b = \begin{pmatrix} -1.5 \\ -1.5 \end{pmatrix}, \quad \Sigma_b = \begin{pmatrix} 1 & 0.3 \\ 0.3 & 1 \end{pmatrix}.$$

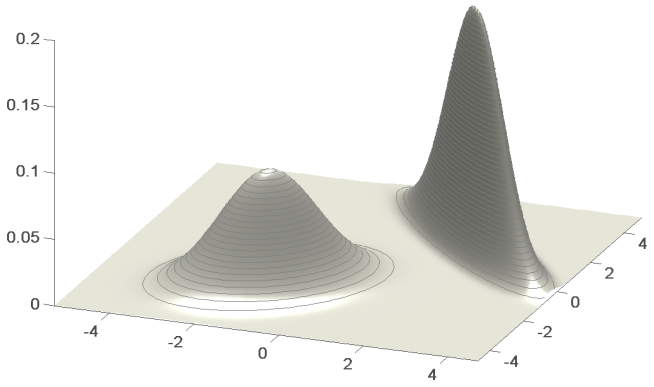


Figure B.2: Density II, $d = 2$.

B.3 Density III

Density III is the standard multivariate Normal density for $x \in \mathbb{R}^d$:

$$f_{III}(x \,|\, \mu, \Sigma, \mu, \Sigma) = \varphi(x \,|\, \mu, \Sigma),$$

where $\varphi(x \,|\, \mu, \Sigma)$ is the multivariate Normal density with mean $\mu \in \mathbb{R}^d$ and $d \times d$ variance-covariance matrix Σ, and

$$\mu = \begin{pmatrix} 0.0 \\ 0.0 \\ \vdots \\ 0.0 \end{pmatrix}, \quad \Sigma = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}.$$

Appendix C

The state space of regular pavings

C.1 Introduction

This appendix discusses the regular paving (RP) state space in terms of the theoretical number of unique regular paving partitions of the same root box and also in terms of the restrictions on that theoretical state space implied by a computer implementation of an RP and, in the case of a statistical regular paving (SRP), the precision of the sample data.

C.2 Catalan numbers and binary trees

Considering RPs as binary trees structures, there are C_k different unique binary trees with $m = |\mathbb{L}(s)| = k+1$ leaves (k 'splits'), where C_k is the Catalan number for k (Equation (3.1)):

$$C_k = \frac{1}{k+1}\binom{2k}{k} = \frac{(2k)!}{(k+1)!(k!)} \quad .$$

The first few values are as follows:

k	0	1	2	3	4	5	6	7	8	9	10
C_k	1	1	2	5	14	42	132	429	1430	4862	16796

Table C.1:

Clearly, the values in the Catalan number sequence get very large.

As $k \to \infty$, $\frac{C_{k+1}}{C_k} = \frac{2(2k+1)}{(k+2)} \to 4$ from below: the number of trees with $k+2$ leaves is up to four times larger than the number of trees with $k+1$ leaves.

There can be more than one way to reach a tree state with $|\mathbb{L}(s)| = m$ leaves from a tree state with $m-1$ leaves. In total there are $k!$ possible routes from the root to *all* the C_k states with $m = k+1$ leaves.[1] This clearly grows much faster than the number of unique states C_k: if the number of leaves increases by 1 from m to $m+1$ the number of states is at most (almost) 4 times the previous number but the number of routes into those states is m times the number of routes into states with m leaves.

Figure C.1 shows the state space for trees with up to 5 leaves ($k = 4$ splits). Each vertex on the graph represents a unique binary tree, identified by its leaf node depth sequence (see Section 3.3.1). Below the label is an expression for the number of routes from the root to that

[1] This can be seen be considering the number of possible 'choices' of nodes to split as the tree size increases. A route from the root (1 leaf) to m leaves has to split at each of the states having leaves 1, 2, ... $m-1$ and at each one there are as many choices of nodes to split to get to the 'next' state (one more leaf than before) as there are leaves, so the total number of splitting sequences to get to m leaves is $1\times2\times\ldots\times(m-2)\times(m-1) = (m-1)! = k!$.

vertex. Adding up the number of routes to all the 4-leaf states gives $6 = k!$ where $k = 4-1 = 3$. Adding up the number of routes to all the 5-leaf states gives $24 = k!$ ($k = 5 - 1 = 4$).

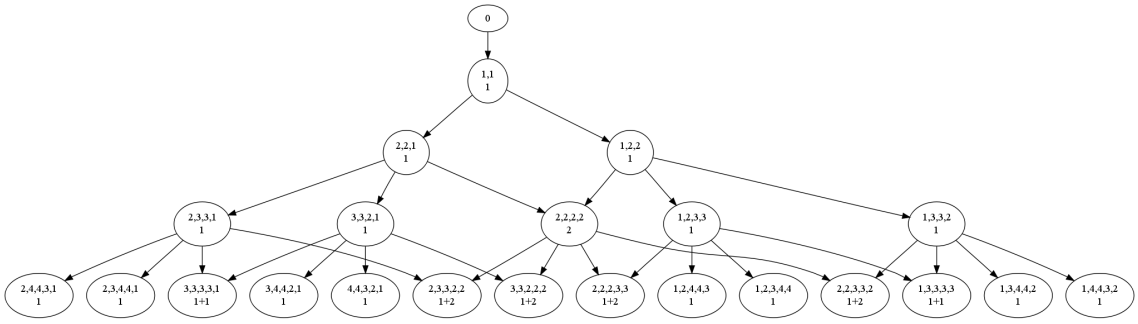


Figure C.1: State space for $m = 5$ leaves ($k = 4$).

Figure C.2 represents the state space for trees with up to one more leaf than in Figure C.1, i.e. up to 6 leaves ($k = 5$ splits). The labels have been omitted. The key point is that most states can be reach via multiple routes from the root.

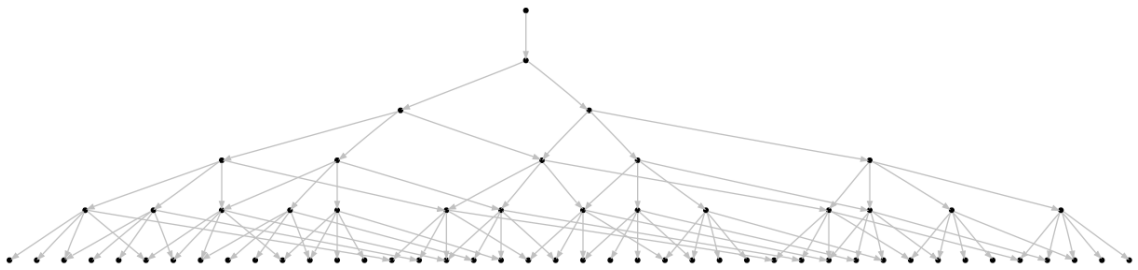


Figure C.2: State space for $m = 6$ leaves ($k = 5$).

For the purposes of analysing algorithms that grow the binary tree by selecting at random a leaf node to be the next to be split, it is useful to note that more routes lead to the more 'balanced' states (i.e., states with most cherry nodes). For example, there are a total of $7! = 5040$ routes from the root to the $C_7 = 429$ states with $m = |\mathbb{L}(s)| = 8$ leaves ($k = 7$), and of these 5040 routes, 80 lead to the completely balanced state '3,3,3,3,3,3,3,3' (all 8 leaves at depth 3). If the binary tree is developing (increasing its total leaves one by one) 'naturally', i.e., just randomly choosing a leaf node to split from its current state and splitting it, then the probability of its 8-leaf state being '3,3,3,3,3,3,3,3' is $\frac{80}{5040} = \frac{1}{63}$. In contrast, there is only one route to each of most unbalanced states (those with two nodes at the maximum possible depth given the number of splits, such as '1,2,3,4,5,6,7,7' or '4,7,7,6,5,3,2,1'). The probability of being in any one of each of these states after 7 such random splits is $\frac{1}{5040}$).

The Catalan numbers can also be built up in a pyramid, the Catalan number for k being derived from the Catalan numbers for $0, 1, 2, \ldots, k-1$ as follows:

$$C_k = \sum_{i=1}^{k} (C_{i-1} C_{k+1-i-1}) = \sum_{i=1}^{k} (C_{i-1} C_{k-i})$$

For example:
$C_2 = C_0C_1 + C_1C_0 = 1 + 1 = 2,$
$C_3 = C_0C_2 + C_1C_1 + C_2C_0 = 2 + 1 + 2 = 5,$
$C_4 = C_0C_3 + C_1C_2 + C_2C_1 + C_3C_0 = 5 + 2 + 2 + 5 = 14,$
$C_5 = C_0C_4 + C_1C_3 + C_2C_2 + C_3C_1 + C_4C_0 = 14 + 5 + 2 \times 2 + 5 + 14 = 42,$
etc.

C.3 Regular pavings represented in a computer

This section describes the limits that will apply to all types of RP when these structures are implemented on a computer. In any computerised implementation of an RP, there will be a limit to how large an RP tree can be held in memory. Each node of the RP is associated with a box; the higher the dimensions of the boxes the smaller the tree that will fit into the available memory. Paradoxically, high-dimensional problems, which generally require larger trees, and where each box consumes more memory, will be more restricted by this than low-dimensional problems.

When a node $\rho\mathsf{v}$ is split the box $\boldsymbol{x}_{\rho\mathsf{v}}$ associated with that node is bisected on its first widest coordinate and the 'bottom' half becomes the left child's box, the 'top' half becomes the right child's box (see Section 3.3). In the computer representation of a box used for this thesis a box $\boldsymbol{x} \in \mathbb{IR}^d$ is an interval vector. The top and bottom of the j^{th} interval represent the upper and lower limits of the box on coordinate j, $j = 1, \ldots, d$. The length of the interval is the size of the box on that dimension. There are two limits on how many times a computer-represented box can be recursively bisected:

- a lower limit on representable total volume in the box; and
- a lower limit on the width of an interval that can be bisected to give distinct 'top' and 'bottom' intervals.

C.3.1 The representation of numbers in computer

Not every real number can be represented accurately in a computer. In double precision format (P754 1985) the normalised representable (absolute) floating point numbers are 0.0 and numbers that can be represented as $2^e \times \left(1 + \sum_{p=1}^{52} b_{[p]}^{-p}\right)$ where

$$e \in \{-1022, -1021, \ldots, -1, 0, 1, \ldots, 1022, 1023\}$$

(i.e., a minimum exponent $e_{min} = -1022$ and maximum exponent $e_{max} = 1023$) and $b_{[p]} \in \{0, 1\}$ (i.e., a binary base), $p = \{1, 2, \ldots, 52\}$ ($52 + 1 = 53$ bits in the mantissa or significand). There is also a sign bit so that any number that can be represented in this way can be either positive or negative.

The smallest normalised (absolute) floating point number using double precision is 1.0×2^{-1022} (P754 1985). Numbers smaller, in absolute terms, than this can be calculated. These are the denormalised numbers. Normalised double-precision numbers have 53 bits in the significand, giving about 15 fractional decimal digits of precision. Denormalised numbers take up to 52 bits out of the significant and move them to the exponent and thus allow a less precise representation of very small numbers. The smallest denormalised floating point number is $1.0 \times 2^{-(1022+52)} = 1.0 \times 2^{-1074} \approx 4.941 \times 10^{-324}$. The purpose of the denormalised numbers is to allow calculations to fail 'gracefully', e.g., to report in some way that an intermediary result is a denormalised number and that further processing cannot be continued. Many arithmetical operations will not deal with denormalised numbers, and of course the precision of the result would be compromised.

If a particular application needs greater precision than that available from the standard double precision format then an extendable or extended precision format could be used. This would move the limit on the smallest representable floating point number but cannot altogether remove it. The implementation of RPs for this thesis uses the standard double precision format described above.

The double precision format means that there are real numbers that cannot be accurately represented. If fact there are whole 'gaps' between the numbers that *can* be represented. These gaps get wider and wider the further away from 0, in an absolute sense, the numbers of interest are. The next number 'up' after 1.0×2^{-1022} is $\left(1.0 + 2^{-52}\right)$: the gap is $2^{-1022-52} = 2^{-1074}$. Between 1.0×2^{-1022} and 1.0×2^{-1021} the gaps are $2^{-1022-52} = 2^{-1074}$. Between 1.0×2^{-1021} and 1.0×2^{-1020} the gaps are 2^{-1073}. Between $1.0 \times 2^{-1} = 0.5$ and $1.0 \times 2^{0} = 1$ the gaps are all $2^{-1-52} = 2^{-53}$. Going to the largest representable numbers, between $2^{1022} \times \left(1 + \sum_{p=1}^{52} 2^{-p}\right)$ and $2^{1023} \times \left(1 + \sum_{p=1}^{52} 2^{-p}\right)$ the gaps are all $2^{1022-52}\left(1 + \sum_{p=1}^{52} 2^{-p}\right) = 2^{970}\left(1 + \sum_{p=1}^{52} 2^{-p}\right) \approx 1.996 \times 10^{292}$.

The larger the gaps, the fewer representable floating point numbers there are in an interval of given length on that region of the real line, or the longer the interval needed to contain the same number of representable floating point numbers. There are $2^{52} - 1$ representable numbers between 0 and 2^{-1022}, between 2^{-1022} and 2^{-1021}, between $2^{-1} = 0.5$ and 1, and between 1 and 2, and between 2 and 4, ..., between 2^{52} and 2^{53}, ...

C.3.2 The volume of a box

The lower limit on the total volume of the box is important because many RP algorithms use the volume of the box and will either fail or give misleading results if that volume cannot be

represented with the expected precision. The implementations of the algorithms discussed in this thesis do not allow a box to be bisected if the volume of the box is less than twice the smallest representable normalised double precision floating number $2 \times 1.0 \times 2^{-1022}$.

A node has depth $\mathsf{d}_{\rho\mathsf{v}} = k$ in the tree if it can be reached by k splits from the root node. If an RP s has root box $\boldsymbol{x}_\rho$ and a node $\rho\mathsf{v}$ in s has depth k, then the volume of the box $\boldsymbol{x}_{\rho\mathsf{v}}$ associated with that node is $\mathrm{vol}\,(\boldsymbol{x}_{\rho\mathsf{v}}) = 2^{-k}\mathrm{vol}\,(\boldsymbol{x}_\rho)$. The restriction on box volume, therefore, translates into a restriction on node depth, and that restriction is a function of the root box volume and the number of dimensions: the smaller the root node volume and the larger the number of dimensions, the more restricted the depth that any node can have.

C.3.3 Bisections of an interval

The gaps between the real numbers that can be represented in a computer described in Section C.3.1 can affect the bisection of a box into two distinct halves. Let the first widest coordinate of a box $\boldsymbol{x}$ be ι. The interval representing the side of the box on coordinate ι is $[\underline{x}, \overline{x}]$. The box should be bisected at the midpoint $\xi = \frac{\underline{x}+\overline{x}}{2}$. The lower box (left child's box) should have interval $[\underline{x}, \xi]$ on coordinate ι and the upper box (right child's box) should have interval $[\xi, \overline{x}]$ on that coordinate However, if there are no representable floating point numbers between $\underline{x}$ and $\overline{x}$, the midpoint will simply be represented one of $\underline{x}$ and $\overline{x}$ themselves. One of the 'child' boxes is therefore the same as the 'parent' box and the other is 'thin' (has no width) on that coordinate. The child boxes, as represented in the computer, are not halves of the parent box. Therefore, the implementations of the algorithms discussed in this thesis do not allow a node $\rho\mathsf{v}$ to be split if the interval of $\boldsymbol{x}_{\rho\mathsf{v}}$ on its first widest coordinate cannot be bisected into two halves both distinct from the interval being bisected, i.e., if there is no representable floating point number between the ends of the interval on the first widest coordinate of $\boldsymbol{x}_{\rho\mathsf{v}}$.

When the boxes to be bisected are multi-dimensional the limits above give the maximum number of times we can split on any single dimension, for example a 2-d box $[0.5, 1]^2$ can be split recursively 2×52 times, corresponding to a maximum node depth of 104. Section C.3.1 shows that the number of representable floating point numbers between the two ends of an interval depends on the magnitude of those end points. For example, there will be no representable floating point number between the end points (place to bisect) of an interval $[2^{52}, 2^{52} + 1]$, whereas there are $2^{52} - 1$ representable numbers between the end points of an interval $[0.5, 1]$. The interval $[2^{52}, 2^{52} + 1]$ cannot be bisected into two halves both distinct from $[2^{52}, 2^{52} + 1]$ but the interval $[0.5, 1]$ can be recursively bisected 52 times before there is no representable midpoint different from the ends left to bisect on. For an interval $[0, 2^{-1}] = [0, 0.5]$ then there can be between 1073 and 53 recursive bisections (more at the, bottom, closest to 0, less at the top).This means that preventing a node $\rho\mathsf{v}$ from being split if the interval of $\boldsymbol{x}_{\rho\mathsf{v}}$ on its first widest coordinate cannot be bisected into two halves both distinct from the interval being

bisected could potentially result in trees with a lopsided shape. With a simple 1-d root box of $[0, 1]$ for example, many more splits will be permitted on the 'left sides', i.e., the nodes with boxes closest to 0 can have more successive descendants than the nodes with boxes closest to 1. If the boxes are multidimensional the same will apply: the 'bottom left' corner of the root box being allowed to become much more finely partitioned than the top right corner.

Effectively, the computer implementation will have the potential to discriminate between different parts of the tree, in terms of what depth of splitting is allowed in that part, on the basis of the position of the boxes associated with that part of the tree within the root box. The larger the root box the more marked the discrimination will be, with areas of the tree associated with boxes closest to 0 being able to split most.

In many cases it may be that, while this is a theoretical possibility, other rules (such as a restriction on the maximum number of leaves or the minimum box volume limit described in Section C.3.2) will restrain at least some asymmetric splitting. The possibility cannot, however, be ignored. In the worst case the restrictions on the state space imposed by the differing bisect-ability of intervals on the far left and right of the root box could result in much deeper splitting on the left than the right and a misleading impression that this reflects features in the sample data rather than just the technicalities of floating point representations in the computer.

The most effective way to prevent this would be to impose a maximum depth $\overline{\mathsf{d}}$ for any leaf node which is at least low enough to ensure that this form of discrimination between different parts of the tree could not occur.

Limiting maximum node depth as well as having maximum number of leaves would have a strong effect on the space of possible tree states. Without depth limits, a tree with $m = |\mathbb{L}(s)| > 1$ leaves can have nodes of depth 1 to $m - 1$. If $\overline{\mathsf{d}}$ is the maximum depth allowed and then there is no permitted tree state with number of leaves $|\mathbb{L}(s)| > 2^{\overline{\mathsf{d}}}$. For example, with no restrictions on the state space at all there are C_{31} unique tree states with 32 leaves. If the maximum depth is 5, there is only one permitted tree state with 32 leaves (the completely balanced tree with all nodes at depth 5),and no permitted tree with 33 leaves or more.

C.4 Statistical regular pavings

Section C.3 discussed the aspects of a computer representation of an RP that effectively restrict the state space of all types of RP. Specific RP types may also have type-related restrictions. Section 4.3.2 (Chapter 4) discusses how the distribution of the sample data over the root box of an SRP $^\circ s$ might affect which nodes in $^\circ s$ SRP are considered to be splittable and hence the space of permitted states for $^\circ s$. A further issue concerning the sample data relates to the precision to which that data is available and is discussed in the following section.

C.4.1 Precision of the data

Section C.3.1 describes the double-precision floating point number representation in a computer (there are 53 bits in the significand, giving about 15 fractional decimal digits of precision). In some situations the data to be analysed using an °*s* may not be available to this level of precision. This would be the case, for example, if the data is read in from a file and was not originally output to that file with the maximum precision allowed by the floating point format.

Limited precision in the data should imply a limit on the maximum depth of the SRP tree so that no node can have a box where the width of that box on any dimension is less than the precision of the data. Taking the 1-d case, if the root box $\boldsymbol{x}_\rho$ has width w then after k recursive splits the width of each of the sub-intervals will be $\frac{w}{2^k}$. If the data only has ϱ fractional decimal digits of precision then the maximum number of splits should be $\max\{k : \frac{w}{2^k} \geq \frac{w}{10^\varrho}\}$, i.e., $k \leq \left(\frac{\varrho \log(10) - \log(w)}{\log(2)}\right)$. For example, if there are only 2 decimal places of precision in the sample data and the root box is $[0, 1]$ then the maximum k is 6; if there are 3 decimal places of precision in the input data then the maximum k is 9. If $\boldsymbol{x}_\rho \in \mathbb{R}^d$ then the limit applies to each coordinate of $\boldsymbol{x}_\rho$ separately.

C.5 Example of a restricted state space

Figure C.3 shows an example of a state space that is a subset of $\mathbb{S}_{0:4}$. Not only is the maximum number of leaves limited to 5, but some additional restriction has also excluded the state labelled '3,3,2,1' from the space. This could be due the issues described in Sections C.3.2 or C.3.3, or, for a space of SRPs, a lower limit $\underline{\#}$ on the number of data points associated with a non-empty node may mean that the right-most node in the state labelled '2,2,1' cannot be split to give '3,3,2,1'.

Note that excluding the state labelled '3,3,2,1' naturally also excludes all the states that may be reached from '3,3,2,1' by further splitting ('3,3,3,3,1', '3,4,4,2,1', '4,4,3,2,1', and '3,3,2,2,1'). Considering Figure C.1 as a directed acyclic graph (i.e., edges between states represent splits), excluding the vertex '3,3,2,1' excludes all the vertices v where there is an out-edge ('3,3,2,1', v).

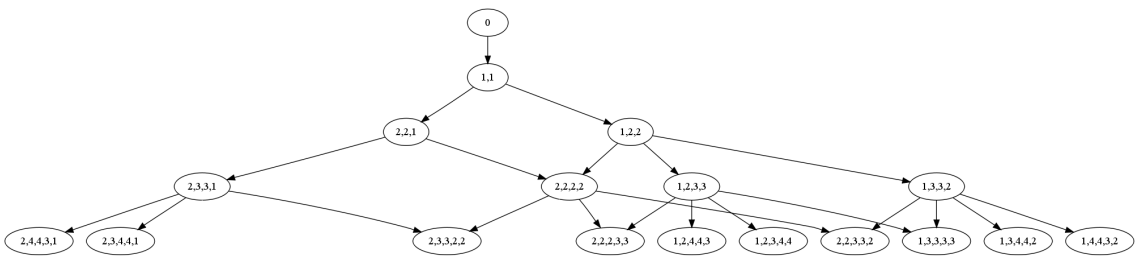


Figure C.3: A restricted state space, a subset of $\mathbb{S}_{0:4}$.

Appendix D

Randomised priority partitioning for SRPs

Randomised priority partitioning requires an appropriate priority function $\psi : \mathbb{L}^{\triangledown}(\,^{\circ}s) \to \mathbb{R}$ on the set of splittable leaf nodes $\mathbb{L}^{\triangledown}(\,^{\circ}s)$ of the statistical regular paving (SRP) $^{\circ}s$. The randomised priority queue operation will split a leaf node that is uniformly chosen at random from $\text{argmax}_{\rho\mathsf{v}\in\mathbb{L}^{\triangledown}(\,^{\circ}s)}\,\psi(\rho\mathsf{v})$, the set of splittable leaf nodes of $^{\circ}s$ which are equally 'large' when measured using ψ. Once the priority function is motivated, a sequential bisection procedure using a *randomised priority queue* over the current set of splittable leaf nodes $\mathbb{L}^{\triangledown}(\,^{\circ}s)$ of the SRP $^{\circ}s$ can be developed.

The RPQ operation `RPQ` shown in Algorithm D.1 allows two forms of 'stopping condition' to be specified. The first is an overall maximum number of leaves in the SRP. This is denoted by $\overline{m}$. The second is a stopping condition that relates to the priority function itself, so that splitting is stopped when the largest 'value' under ψ of any splittable leaf node is less than or equal to a specified value $\overline{\psi}$, i.e., when $\max\{\psi(\rho\mathsf{v}) : \rho\mathsf{v} \in \mathbb{L}^{\triangledown}(\,^{\circ}s)\} \leq \overline{\psi}$. These are the only two forms of stopping that can be guaranteed to be effective. In addition, a priority queue partitioning operation must cease if there are no more splittable leaf nodes ($\mathbb{L}^{\triangledown}(\,^{\circ}s) = \emptyset$)

The result will be a SRP where $\mathbb{L}^{\triangledown}(\,^{\circ}s) = \emptyset$ or $\psi(\rho\mathsf{v}) \leq \overline{\psi}\ \forall \rho\mathsf{v} \in \mathbb{L}^{\triangledown}(\,^{\circ}s)$ or $|\mathbb{L}(\,^{\circ}s)| \leq \overline{m}$. Clearly, in general it cannot be guaranteed that $\psi(\rho\mathsf{v}) \leq \overline{\psi}\ \forall \rho\mathsf{v} \in \mathbb{L}^{\triangledown}(\,^{\circ}s)$ *and* $|\mathbb{L}(\,^{\circ}s)| \leq \overline{m}$.

Algorithm D.1: `RPQ`$(\,^{\circ}s, \psi, \overline{\psi}, \overline{m})$

---

input : SRP $^{\circ}s$ with root box $\boldsymbol{x}_{\rho}$,
priority function $\psi : \mathbb{L}^{\triangledown}(\,^{\circ}s) \to \mathbb{R}$,
$\overline{\psi}$ the maximum value of $\psi(\rho\mathsf{v}) \in \mathbb{L}^{\triangledown}(\,^{\circ}s)$ for any splittable leaf node in the final SRP,
$\overline{m}$ the maximum number of leaves in the final SRP.

output : $^{\circ}s$ such that $\mathbb{L}^{\triangledown}(\,^{\circ}s) = \emptyset$ or $\psi(\rho\mathsf{v}) \leq \overline{\psi}\ \forall \rho\mathsf{v} \in \mathbb{L}^{\triangledown}(\,^{\circ}s)$ or $|\mathbb{L}(\,^{\circ}s)| \leq \overline{m}$.

while $\mathbb{L}^{\triangledown}(\,^{\circ}s) \neq \emptyset\ \ \&\ \ |\mathbb{L}(\,^{\circ}s)| < \overline{m}\ \ \&\ \ \psi\left(\text{argmax}_{\rho\mathsf{v}\in\mathbb{L}^{\triangledown}(\,^{\circ}s)}\,\psi(\rho\mathsf{v})\right) > \overline{\psi}$ **do**

  $\rho\mathsf{v} \leftarrow$ `random_sample` $\left(\underset{\rho\mathsf{v}\in\mathbb{L}^{\triangledown}(\,^{\circ}s)}{\text{argmax}}\ \psi(\rho\mathsf{v})\right)$

  Split $\rho\mathsf{v}$: $\triangledown_{\text{SRP}}(\rho\mathsf{v}) = \{\rho\mathsf{v}\mathsf{L}, \rho\mathsf{v}\mathsf{R}\}$ `// split the sampled node`

  `RPQ`$(\,^{\circ}s, \psi, \overline{\psi}, \overline{m})$

end

The statistically equivalent block (SEB)-based priority function is $\psi(\rho\mathsf{v}) = \#\boldsymbol{x}_{\rho\mathsf{v}}$, i.e., the number of sample points associated with a node $\rho\mathsf{v}$. This ψ prioritises the splitting of leaf nodes with the largest numbers of data points associated with them. The priority function-related stopping condition is $\overline{\psi} = \overline{\#}$,

At the end of the operation, the SRP $^{\circ}s$ will be such that either $\mathbb{L}^{\triangledown}(\,^{\circ}s) = \emptyset$ or $|\mathbb{L}(\,^{\circ}s)| \leq \overline{m}$

or $\#\boldsymbol{x}_{\rho\mathsf{v}} \leq \overline{\#}\ \forall \rho\mathsf{v} \in \mathbb{L}^{\triangledown}(\,^{\circ}s)$. The operation may only be considered to be successful if $|\mathbb{L}(\,^{\circ}s)| \leq \overline{m}$ and $\#\boldsymbol{x}_{\rho\mathsf{v}} \leq \overline{\#}\ \forall \rho\mathsf{v} \in \mathbb{L}^{\triangledown}(\,^{\circ}s)$ (as noted above, this is not guaranteed).

Appendix E

The Metropolis-Hastings MCMC sampler

E.1 Introduction

This appendix gives a detailed analysis of the behaviour of the Metropolis-Hastings Markov chain Monte Carlo (MCMC) sampler described in Section 6.4.

E.2 The stay-split-merge base chain

Under the stay-split-merge base Markov chain $\{Y(t)\}_{t\in\mathbb{Z}_+}$ on the finite state space ${}^{\circ}\tilde{\mathbb{S}}$ the transition probabilities between any two states ${}^{\circ}s,\, {}^{\circ}s' \in {}^{\circ}\tilde{\mathbb{S}}$ (see Equation (6.4)) are:

$$Q({}^{\circ}s,\, {}^{\circ}s') = \begin{cases} \dfrac{1-\varsigma}{2|\mathbb{L}^{\triangledown}({}^{\circ}s)|} & \text{if a node } \rho\mathsf{v} \in \mathbb{L}^{\triangledown}({}^{\circ}s) \text{ can be split once to get } {}^{\circ}s' \\ \dfrac{1-\varsigma}{2|\mathbb{C}({}^{\circ}s)|} & \text{if a node } \rho\mathsf{v} \in \mathbb{C}({}^{\circ}s) \text{ } {}^{\circ}s \text{ can be reunited once to get } {}^{\circ}s' \\ \varsigma & \text{if } {}^{\circ}s = {}^{\circ}s' \\ 0 & \text{otherwise .} \end{cases}$$

If the current state ${}^{\circ}s$ has only the root node ρ (no partitioning, and therefore no cherry nodes) then with probability $\frac{1-\varsigma}{2}$ the base chain will give a proposal to split the root and with probability $\frac{1+\varsigma}{2}$ no move will be proposed. In general the proposal state ${}^{\circ}s'$ may be the current state ${}^{\circ}s$, or ${}^{\circ}s$ with one cherry $\rho\mathsf{v} \in \mathbb{C}({}^{\circ}s)$ merged, or ${}^{\circ}s$ with one leaf node $\rho\mathsf{v} \in \mathbb{L}^{\triangledown}({}^{\circ}s)$ split.

E.3 The acceptance probability

If the proposal state ${}^{\circ}s'$ is not the current state ${}^{\circ}s$ then the acceptance probability (Section 6.4.2) is

$$a({}^{\circ}s,\, {}^{\circ}s') := \min\left\{1, \frac{\pi({}^{\circ}s')Q({}^{\circ}s',\, {}^{\circ}s)}{\pi({}^{\circ}s)Q({}^{\circ}s,\, {}^{\circ}s')}\right\} \ ,$$

and

$$\frac{\pi({}^{\circ}s')Q({}^{\circ}s',\, {}^{\circ}s)}{\pi({}^{\circ}s)Q({}^{\circ}s,\, {}^{\circ}s')} = \frac{\mathcal{L}\left(\hat{f}'_n\right)}{\mathcal{L}\left(\hat{f}_n\right)} \frac{\Pr\{{}^{\circ}s'\}}{\Pr\{{}^{\circ}s\}} \frac{Q({}^{\circ}s',\, {}^{\circ}s)}{Q({}^{\circ}s,\, {}^{\circ}s')} \ ,$$

where $\hat{f}'_n$ and $\hat{f}_n$ are the histograms based on the partitions of ${}^\circ s'$ and ${}^\circ s$, respectively. In log terms,

$$\log\left(\frac{\pi({}^\circ s')Q({}^\circ s',\,{}^\circ s)}{\pi({}^\circ s)Q({}^\circ s,\,{}^\circ s')}\right) = \log\left(\frac{\mathcal{L}\left(\hat{f}'_n\right)}{\mathcal{L}\left(\hat{f}_n\right)}\right) + \log\left(\frac{\Pr\{{}^\circ s'\}}{\Pr\{{}^\circ s\}}\right) + \log\left(\frac{Q({}^\circ s',\,{}^\circ s)}{Q({}^\circ s,\,{}^\circ s')}\right) \quad .$$

E.3.1 Influence of the proposal distribution on the probability of accepting the proposal

The ratio $\frac{Q({}^\circ s',\,{}^\circ s)}{Q({}^\circ s,\,{}^\circ s')}$ is a function of the number of cherries and splittable leaves in the current state and proposed state.

Taking the cherries first, let c be the number of cherries in the current state ${}^\circ s$ and let c' be the number of cherries in the proposed state ${}^\circ s'$. If the proposed state ${}^\circ s'$ is achieved by splitting one of the leaves of ${}^\circ s$, then

$$c' = \begin{cases} c+1 & \text{if the node that is to be split does not have a sibling leaf;} \\ c & \text{if the node that is to be split has a sibling leaf.} \end{cases}$$

If the proposal ${}^\circ s'$ is achieved by merging a pair of sibling child leaves of ${}^\circ s$ back into their parent node, then

$$c' = \begin{cases} c & \text{if the new leaf node formed by the merge would have a sibling leaf;} \\ c-1 & \text{if the new leaf node formed by the merge would not have a sibling leaf.} \end{cases}$$

Now let m be the number of *splittable* leaves in the current state ${}^\circ s$ and let l' be the number of splittable leaves in the proposed state ${}^\circ s'$. If a split is proposed then each splittable leaf is equally likely to be the one split. Splitting a leaf clearly means that that leaf node goes out of the set of splittable leaf nodes, but can result in either 2 or 1 or 0 new splittable child nodes being added to that count (2 if both new child nodes are splittable, 1 if only one is splittable, 0 if neither are).

$$l' = \begin{cases} l+1 & \text{if the node that is to be split will have two splittable child nodes;} \\ l & \text{if the node that is to be split will have only one splittable child node;} \\ l-1 & \text{if the node that is to be split will have no splittable child nodes.} \end{cases}$$

Similarly, for a merge proposal, a merge will form a new splittable leaf node and the two erstwhile child leaf nodes cease to exist, but it is not necessarily the case that both were *splittable* leaf nodes. A merge can therefore result in a loss of either 2 or 1 or 0 *splittable* leaf

nodes (2 if both child leaves were splittable, 1 if only one was splittable, 0 if neither were).

$$l' = \begin{cases} l+1 & \text{if the node that is to be merged has no splittable child nodes;} \\ l & \text{if the node that is to be merged has one splittable child node;} \\ l-1 & \text{if the node that is to be merged has two splittable child nodes.} \end{cases}$$

If the proposal is a split then

$$\frac{Q(\circ s', \circ s)}{Q(\circ s, \circ s')} = \frac{l}{c'} \ ,$$

and $\frac{l}{c'} \geq 1$ unless the number of splittable leaf nodes is very much lower than the total actual number of leaf nodes. If all leaf nodes are splittable then $l > c$ (either $l = 1$ and $c = 0$ or at a minimum, $l = 2c$) and at most $c' = c + 1$. Then the only state for which $\frac{l}{c'} = 1$ is when there is a single leaf node (the root). Otherwise the ratio is usually > 1 unless splittable leaf nodes are only a low proportion of the total leaf nodes. The ratio of the proposal probabilities within the acceptance probability will therefore tend to increase the acceptance probability of a proposal to split a leaf node.

If the proposal is a merge then

$$\frac{Q(\circ s', \circ s)}{Q(\circ s, \circ s')} = \frac{c}{l'} \ ,$$

and $\frac{c}{l'} \leq 1$, unless the number of splittable leaf nodes is very much lower than the total actual number of leaf nodes. If all leaf nodes are splittable then $l > c$ (as above) and at least $l' = l - 1$. Then the only situation in which $\frac{c}{l'} = 1$ is when there are two leaf nodes and one cherry. Otherwise the ratio is strictly < 1. The ratio of the proposal probabilities within the acceptance probability will therefore tend to lower the acceptance probability of a proposal to merge a cherry node.

Not only does the influence of the proposal chain on the acceptance probabilities tend to encourage splitting, but the more 'unbalanced' a tree is (the higher the ratio of leaf nodes to cherries), the stronger this influence is and the more further splitting is encouraged. Furthermore, if an SRP tree has already become quite unbalanced (with more partitioning in one large branch than in the rest of the tree), then the leaf nodes from this part of the tree may comprise a large proportion of the total splittable leaf nodes. In this situation, there is a commensurately large probability that, if a split is proposed, it is be one of the leaf nodes from the already most-split branch that is selected to be split and hence that the further splitting will occur in the already most split branch.

E.3.2 Influence of the likelihood on the probability of accepting the proposal

For a split, the likelihood ratio component of the acceptance probability is:

$$\frac{\mathcal{L}\left(\hat{f}'_n\right)}{\mathcal{L}\left(\hat{f}_n\right)} = 2^{\#\boldsymbol{x}_{\rho\mathsf{v}}} \frac{(\#\boldsymbol{x}_{\rho\mathsf{v}\mathsf{L}})^{\#\boldsymbol{x}_{\rho\mathsf{v}\mathsf{L}}} (\#\boldsymbol{x}_{\rho\mathsf{v}\mathsf{R}})^{\#\boldsymbol{x}_{\rho\mathsf{v}\mathsf{R}}}}{(\#\boldsymbol{x}_{\rho\mathsf{v}})^{\#\boldsymbol{x}_{\rho\mathsf{v}}}} \geq 1 \ ,$$

and for a merge, the likelihood ratio is:

$$\frac{\mathcal{L}\left(\hat{f}'_n\right)}{\mathcal{L}\left(\hat{f}_n\right)} = \frac{1}{2}^{\#\boldsymbol{x}_{\rho\mathsf{v}}} \frac{(\#\boldsymbol{x}_{\rho\mathsf{v}})^{\#\boldsymbol{x}_{\rho\mathsf{v}}}}{(\#\boldsymbol{x}_{\rho\mathsf{v}\mathsf{L}})^{\#\boldsymbol{x}_{\rho\mathsf{v}\mathsf{L}}} (\#\boldsymbol{x}_{\rho\mathsf{v}\mathsf{R}})^{\#\boldsymbol{x}_{\rho\mathsf{v}\mathsf{R}}}} \leq 1$$

(see Equation (3.4)).

In the most balanced cases (for a split, where exactly half the data points associated with the node would go to each child, or for a merge where each child has exactly half the data points associated with the parent) the likelihood ratio is 1 and it has no influence on the probability of accepting a proposal to merge or split.

In the most unbalanced cases (for a split, where all $\#\boldsymbol{x}_{\rho\mathsf{v}}$ data points of the proposed node $\rho\mathsf{v}$ would go to just one child, or for a merge where one child has all the data and the other none):

- For a proposal to split node $\rho\mathsf{v}$, $\frac{\mathcal{L}\left(\hat{f}'_n\right)}{\mathcal{L}\left(\hat{f}_n\right)} = 2^{\#\boldsymbol{x}_{\rho\mathsf{v}}}$.
- For a proposal to merge node $\rho\mathsf{v}$, $\frac{\mathcal{L}\left(\hat{f}'_n\right)}{\mathcal{L}\left(\hat{f}_n\right)} = \left(\frac{1}{2}\right)^{\#\boldsymbol{x}_{\rho\mathsf{v}}}$

E.3.3 Influence of the prior on the probability of accepting the proposal

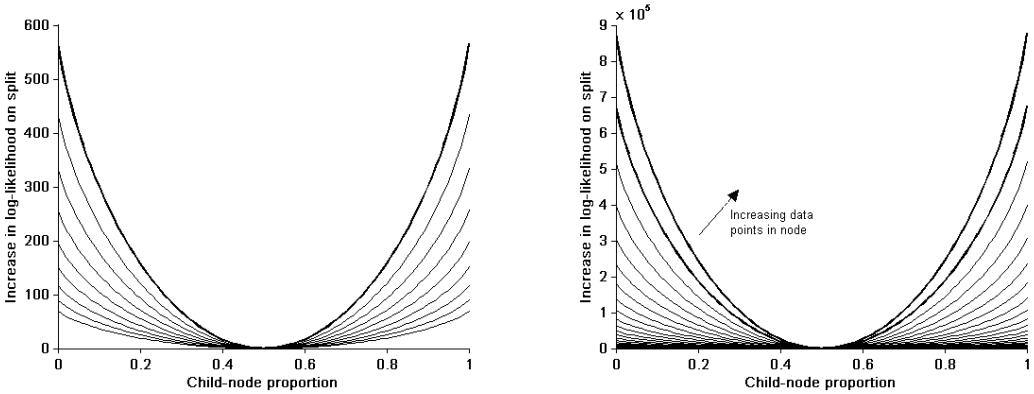
The prior discussed here is the natural Catalan prior described in Section 6.2. For a proposal to split, with current state $^\circ s$ with $m = |\mathbb{L}(^\circ s)|$ leaves and $k = m - 1$ splits, the prior ratio in the acceptance probability calculation for a proposal to move to a state $^\circ s'$ with $m + 1$ leaves and $k = m$ splits is

$$\left(\frac{\Pr\{^\circ s'\}}{\Pr\{^\circ s\}}\right) = \left(\frac{C_k}{C_{k+1}}\right)^2 = \left(\frac{(k+2)}{2(2k+1)}\right)^2 \ .$$

This ratio is $1^2 = 1$ for a proposal to split the root ($m = 1$), i.e., the prior is neutral in its influence on the acceptance probability for a proposal to split the root box. For the next split the ratio is $\left(\frac{1}{2}\right)^2$. As the number of existing splits k increases, this ratio falls. As $k \to \infty$, $\left(\frac{C_k}{C_{k+1}}\right)^2 = \left(\frac{(k+2)}{2(2k+1)}\right)^2 \to \left(\frac{1}{4}\right)^2 = \left(\frac{1}{2}\right)^4$ from above.

E.3.4 Influence of $\#\boldsymbol{x}_{\rho\mathsf{v}}$ on the probability of accepting the proposal

The larger the number of data points $\#\boldsymbol{x}_{\rho\mathsf{v}}$ associated with a node, the more influence an asymmetry or imbalance in the proportion of $\#\boldsymbol{x}_{\rho\mathsf{v}}$ that would be associated with each of the two children has on the acceptance probability, through its effect on the likelihood ratio. Figure E.1 illustrates this by plotting $\log\left(2^{\#\boldsymbol{x}_{\rho\mathsf{v}}}\frac{(p\#\boldsymbol{x}_{\rho\mathsf{v}})^{p\#\boldsymbol{x}_{\rho\mathsf{v}}}((1-p)\#\boldsymbol{x}_{\rho\mathsf{v}})^{(1-p)\#\boldsymbol{x}_{\rho\mathsf{v}}}}{(\#\boldsymbol{x}_{\rho\mathsf{v}})^{\#\boldsymbol{x}_{\rho\mathsf{v}}}}\right)$ against $p = \frac{\#\boldsymbol{x}_{\rho\mathsf{vL}}}{\#\boldsymbol{x}_{\rho\mathsf{v}}}$ for various levels of $\#\boldsymbol{x}_{\rho\mathsf{v}}$. Figure E.1(a) shows plots for $100 \leq \#\boldsymbol{x}_{\rho\mathsf{v}} \leq 1,000$ (the order of magnitude of $\#\boldsymbol{x}_{\rho\mathsf{v}}$ that might be associated with a leaf node of an SRP at a reasonable depth in the tree if the total number of data points associated with the whole SRP is reasonably large, say 10,000–1,000,000). The higher curves are for the larger values of $\#\boldsymbol{x}_{\rho\mathsf{v}}$. Figure E.1(b) shows plots for $1,000 \leq \#\boldsymbol{x}_{\rho\mathsf{v}} \leq 1,264,600$, the larger values being the order of magnitude of $\#\boldsymbol{x}_{\rho\mathsf{v}}$ that might be associated with a leaf node at low depth in the tree (possibly the root node) if the total number of data points associated with the whole SRP is again reasonably large. Again the higher curves relate to the largest values of $\#\boldsymbol{x}_{\rho\mathsf{v}}$.



(a) $100 \leq \#\boldsymbol{x}_{\rho\mathsf{v}} \leq 1000$. (b) $1000 \leq \#\boldsymbol{x}_{\rho\mathsf{v}} \leq 1,264,600$.

Figure E.1: The effect of $\#\boldsymbol{x}_{\rho\mathsf{v}}$ on the log of the likelihood ratio.

For example, if a node $\rho\mathsf{v}$ selected for a split has $\boldsymbol{x}_{\rho\mathsf{v}} = 100$ data points associated with it then, if either child would be associated with less than 39% of these 100 data points, $\frac{\mathcal{L}(\hat{f}_n')}{\mathcal{L}(\hat{f}_n)} > \left(\frac{1}{4}\right)^2$ and the acceptance probability will be 1 irrespective of how split the SRP is already (unless the number of splittable leaf nodes is very much lower than the total actual number of leaf nodes). The more data points there are associated with a node the more 'sensitive' the acceptance probability will be to relatively small imbalances in the proportions that would be associated with the child nodes. If a node $\rho\mathsf{v}$ selected for a split has $\boldsymbol{x}_{\rho\mathsf{v}} = 1,000$ data points associated with it then if either child would be associated with less than 46.3% of these 1000 data points, $\frac{\mathcal{L}(\hat{f}_n')}{\mathcal{L}(\hat{f}_n)} > \left(\frac{1}{4}\right)^2$ and the acceptance probability will be 1. For a node with 5,000 data points the 'unbalance' proportion is about 48.3%.

E.3.5 Summary of the influences on the acceptance probability using the natural Catalan prior

This section summarises the influences on the probability of accepting a proposal from the stay-split-merge base chain (Section 6.4.1) when the natural Catalan prior (Section 6.2) is used.

If an SRP has only one node, the root node, the probability of a split proposal is $\frac{1-\varsigma}{2}$ and, once received, the proposal will always be accepted because $\frac{\mathcal{L}(\hat{f}'_n)}{\mathcal{L}(\hat{f}_n)} \geq 1$, $\frac{\Pr\{\circ s'\}}{\Pr\{\circ s\}} = \left(\frac{C_0}{C_1}\right)^2 = 1$, $\frac{Q(\circ s', \circ s)}{Q(\circ s, \circ s')} = \frac{l}{c'} = 1$ and so the acceptance probability for such a proposal will be 1. A proposal to remerge the resulting two leaf nodes back into the root will have an acceptance probability that depends only on the ratio of the likelihoods $\frac{\mathcal{L}(\hat{f}'_n)}{\mathcal{L}(\hat{f}_n)}$. Unless the root node data is absolutely evenly distributed between the two child nodes this will be less than one and a re-merge proposal may or may not be accepted

Other than in the case of a split of the root node, the influence of the proposal in the acceptance probability is generally to encourage splitting. The prior will act against this (discourage splitting) but the ratio of the prior probabilities $\frac{\Pr\{\circ s'\}}{\Pr\{\circ s\}}$ in the acceptance probability calculation for a split proposal is bounded from below by $\left(\frac{1}{4}\right)^2$. Figure E.1 shows how easily this can be dominated by the effect of the likelihood term $\frac{\mathcal{L}(\hat{f}'_n)}{\mathcal{L}(\hat{f}_n)}$ in the acceptance probability, and how the strength of the influence of the likelihood increases with the number of data points associated with the node.

A proposal to split a node $\rho\mathsf{v}$ when all the data associated with that node would go to just one of its children will have acceptance probability 1 if $\boldsymbol{x}_{\rho\mathsf{v}} \geq 4$ (because $\left(\frac{C_k}{C_{k+1}}\right)^2 < 2^{\#\boldsymbol{x}_{\rho\mathsf{v}}}$ $\forall$ $k \geq 0$, $\boldsymbol{x}_{\rho\mathsf{v}} \geq 4$), but if $\boldsymbol{x}_{\rho\mathsf{v}} < 4$ the acceptance probability may be less than 1 (depending on the influence of the proposal distribution and the number of existing splits k).

Overall, the behaviour of the acceptance probability is in many ways very suitable for data-adaptive partitioning of an SRP to form a density estimate: nodes with most data associated with them are more sensitive to asymmetries (imbalances of that data between their prospective child nodes), while nodes with fewer data points are more easily restrained from splitting by the influence of the prior, and when the number of data points associated with a node is very low the prior may be able to dominate the influence of the likelihood even if all the data points associated with a node would go to just one child.

A disadvantage is that the acceptance probability is sensitive (through the proposal distribution) to the ratio of leaves to cherries in the SRP in a way that tends to encourage more splitting of an unbalanced tree. In addition, the acceptance probability calculation for splits of the root node or merges of a two-leaf node state back to the root node depends entirely on the likelihood ratio. With sample data perfectly evenly distributed between the child nodes of the root node, proposals both to split the root and merge the resulting cherry will always be accepted. With not-quite evenly distributed data proposals to split the root will always be

accepted but a subsequent merge proposal may also be accepted and so the chain may move back and forth between one-leaf and two-leaf states for protracted periods.

E.4 The choice of prior

The discussion above shows that the influence of the prior on the behaviour of the Metropolis-Hastings Markov chain depends on the magnitude of the total sample size (the number of data points $n = \#\boldsymbol{x}_\rho$ associated with the root node ρ of the SRP). As n increases the natural Catalan prior (Section 6.2) becomes relatively weaker as the effect of the likelihood on the acceptance probability becomes stronger (see Section E.3.2). In general the chain will explore more deeply-split states when n is larger.

An alternative to the natural Catalan prior described in this thesis would be some form of 'tunable' prior, for example, a prior selected from a family of priors indexed by a temperature parameter. A 'hotter' prior would be weaker and would allow the chain to explore deeper (more split) states; a colder prior would have a stronger effect discouraging splitting. The temperature could then be set in relation to the sample size n. The issue with this is that, once again, it is difficult to select the correct temperature when the true distribution of the data is unknown and it is therefore impossible to know what depth of tree (temperature of prior) is appropriate.

E.5 Why do the chains get trapped?

Given all the factors discussed above it may seem odd that chains created using the stay-split-merge Metropolis-Hastings MCMC algorithm described here are prone to getting trapped in a sub-space of the total available state space for long periods. The reason is that the likelihood term in the acceptance probability depends only on the effect of the *immediately proposed transition*, not on the *distribution of the data* (see Section 4.3.3).

The most obvious manifestation of a trapped chain is when the states in a chain stay close to the root node state (little partitioning of the root box or of a very large part of the root box) for a long period. This is most likely to happen when the sample data appears to be evenly distributed for several successive bisections. For example, a chain using an SRP with a root box that is symmetric about the origin and multivariate Gaussian data will typically take some time to move away from shallowly-partitioned states if started from an initial state with just the root node. The higher the number of dimensions in the Gaussian data, the deeper the tree has to get before the sample data associated with a node does not appear to be close-to evenly distributed between its (prospective) child nodes. In this situation proposals to split a node can still have reasonably high acceptance probabilities (the amount of data associated with the nodes close to the root is high and the number of existing splits is low), but a subsequent proposal to merge the split node may well also be made and also accepted.

The chain is not trapped in the sense of not accepting any move, but it is prone to move back and forth in a small number of low-leaf states, making and then reversing the same changes.

A variation on this occurs when a large branch of the tree manages to achieve much deeper splitting but the rest stays trapped in a small number of states as described above. The result can be a much more uneven partitioning than is warranted by the distribution of the sample data. As is noted above, once one major branch (large area of the root box) is considerably more deeply split than another and the leaf nodes from this already well-split branch form a high proportion of the total splittable leaf nodes, then the uneven partitioning can persist for a long time.

The effect on the chain can be slightly different when a node deeper in the tree has apparently evenly distributed data (but with hidden asymmetries) associated with it, compared with the situation discussed above in relation to nodes close to the root. The acceptance probability for a split proposal may be quite low (a node deeper in the tree will typically have less data associated with it, and the number of existing splits in the whole tree is higher). In this case it may take some time for a split proposal, once made, to be accepted at all.

This form of 'locally-trapped' state is very much less obvious because the tree as a whole will still be changing state as other leaf or cherry nodes are proposed for, and accept, splits or merges. It can be most easily diagnosed by comparing the behaviour of different chains. In some chains the crucial proposal may be accepted and the SRP is able to enter a whole new group of states that are not being explored in the locally-trapped chain.

Appendix F

Finding multiple initial states for an SRP MCMC process

F.1 Introduction

This appendix gives an outline of the heuristic method used in this thesis to get c states from an SRP ${}^{\circ}s \in {}^{\circ}\mathbb{S}_0$ (an SRP with just the root node) to act as initial states for a subsequent multiple-chain MCMC process.

F.2 Outline

SEB RPQ. Start with an SRP ${}^{\circ}s \in {}^{\circ}\mathbb{S}_0$ and get a sequence of states $\{S(k)\}_{k\in\mathbb{Z}_+}$ where state $S(k)$ has leaves $|\mathbb{L}(S(k))| = k+1$ using the `RPQ`$({}^{\circ}s, \psi, \overline{\psi}, \overline{m})$ procedure (Algorithm D.1 in Appendix D) and the SEB priority function (see Equation (5.1)) for ψ, or with a short 'carving' RPQ (see Equation (5.2)) followed by the SEB RPQ.

Find the state with the highest log-posterior. Find the maximum log-posterior state ${}^{\circ}s^{\star} = S(k^{\star})$ in $\{S(k)\}_{k\in\mathbb{Z}_+}$ such that

$$k^{\star} = \underset{k}{\operatorname{argmax}}\{\log \pi\left(S(k)\right) \,:\, S(k) \in \{S(k)\}_{k\in\mathbb{Z}_+}\}.$$

Find an earlier state with a fairly high log-posterior. Find the first state $S(\underline{k})$ in $\{S(k)\}$ with log-posterior at least some fraction α of the maximum log-posterior $\log \pi\left({}^{\circ}s^{\star}\right)$, i.e., find $S(\underline{k})$ such that

$$\underline{k} = \underset{k}{\operatorname{argmin}}\{\log \pi\left(S(k)\right) \,:\, S(k) \in \{S(k)\}_{k\in\{1,\ldots,k^{\star}\}} \text{ and } \log \pi\left(S(k)\right) \geq \alpha \log \pi\left({}^{\circ}s^{\star}\right)\}.$$

$\alpha = 0.95$ is used in all the examples shown in this thesis unless stated otherwise.

Select c initial states spread around ${}^{\circ}s^{\star}$. Select c well-spaced initial states from the sub-sequence $\{S(k)\}_{k\in\{\underline{k},\ldots,\overline{k}=k^{\star}+(k^{\star}-\underline{k})\}}$. In the implementation used in this thesis the selection always includes the maximum log-posterior state ${}^{\circ}s^{\star} = S(k^{\star})$. $\lceil\frac{c-1}{2}\rceil$ states are selected between states $S(\underline{k})$ and $S(k^{\star})$ (including $S(\underline{k})$ but not including $S(k^{\star})$) so that these states and $S(k^{\star})$ are approximately equi-distant apart in the sub-sequence $\{S(k)\}_{k\in\{\underline{k},\ldots,k^{\star}\}}$. Similarly $c - \lceil\frac{c-1}{2}\rceil$ states are selected between states $S(k^{\star})$ and $S(\overline{k})$ so that $S(k^{\star})$ is included in the selection and these states are approximately equi-distant apart in the sub-sequence $\{S(k)\}_{k\in\{k^{\star},\ldots,\overline{k}\}}$.

F.3 Discussion

The outcome of this process is a set of c initial states for a multiple-chain MCMC process. These initial states will depend on the values of the parameters used for the SEB RPQ process: the maximum number of leaves $\overline{m}^{(S)}$ and the value of $\overline{\psi}^{(S)}$. In the testing carried out in the course of this thesis the final estimate of the posterior expectation of the distribution of RMRP histograms has been found to be fairly robust to the exact values of these parameters provided that the values of $\overline{\psi}^{(S)}$ and $\overline{m}^{(S)}$ allow the SEB RPQ to run for long enough to capture the large initial changes in the log-posterior mass described in Section 6.6.3.

Figure 6.7(a) in Chapter 6 illustrates the use of the heuristic in a typical situation, showing the trace of the (unnormalised) log-posterior over the whole SEB RPQ sequence, the point where the maximum log-posterior mass is attained, and the sub-sequence of states from which multiple initial states are chosen ("the selection region").

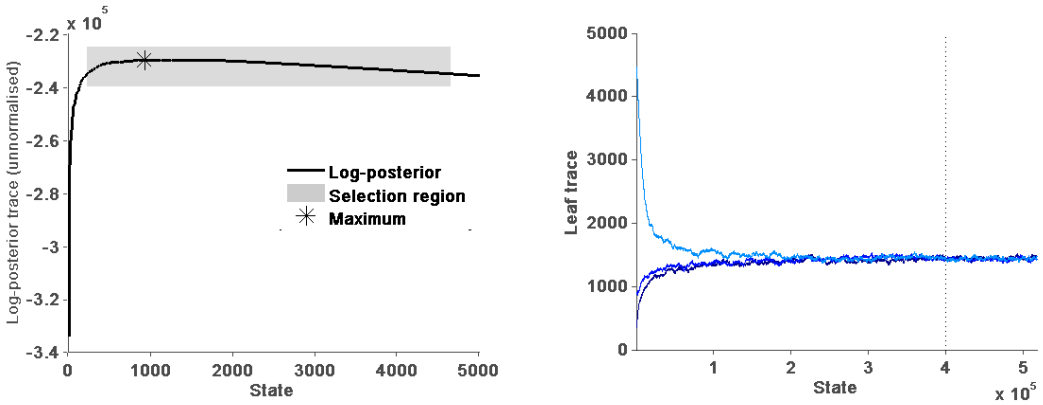
As is discussed above, the RPQ is intentionally run for a relatively long time to ensure that it includes the maximum log-posterior point. Although the range of numbers of leaves in the states covered by the selection region shown in Figure 6.7(a) appears to be limited in relation to the largest number of leaves in the RPQ sequence, the results of testing the heuristic suggest that the selection region described here still provides a reasonably well-diversified range of initial states for a subsequent MCMC process.

A variation on the method described above would be to extend the selection region further, to the state where, say, the log-posterior falls below α of the maximum log-posterior for the first time. Results using various different densities during the course of the research for this thesis suggest that this does not give a better final outcome. The number of leaves in the states sampled from the chains once convergence has been achieved is typically within the range covered by the selection region as described above (see Figure 6.7(b)), and the selection region identified by the method described in Section F.2 is usually broad enough to protect against convergence being falsely diagnosed simply because of the similarity of initial states.

Figure F.1(a) shows the selection region from Figure 6.7(a) (Chapter 6) extended, as described above, to state where the log-posterior falls below α of the maximum log-posterior for the first time. Figure F.1(b) shows the leaf traces from three chains started at the first state, maximum log-posterior point, and last state in this extended region. It takes longer for the more widely dispersed chains to converge but the leaf traces eventually settle down as in Figure 6.7(b) and there is little change in the final outcome.

However, there might be situations in which the heuristic above does not give well-enough diversified initial states. In this case, trace plots of the different chains such as Figure 6.7(b) would show the chains moving very closely together even in their early states. The process can then be re-run using a lower value of α to get a larger selection region for the initial states.

The SEB RPQ is computationally efficient (fast) and so a conservatively large value (relative to the sample size n) of $\overline{m}^{(S)}$ and small value of $\overline{\psi}^{(S)}$ can be used. If a carving phase is included



(a) Extended selection region.

(b) Leaf trace for three chains.

Figure F.1: Selecting multiple initial states with extended selection region.

in the process (See Section 5.4) then $\overline{\psi}^{(\mathrm{C})}$ can be set to 0.0, to ensure that this phase runs until the number of leaves in the SRP is the maximum number of leaves $\overline{m}^{(\mathrm{C})}$ specified for the carving RPQ. The results of testing using different values for $\overline{m}^{(\mathrm{C})}$ suggest that the final outcome is typically more sensitive to this than to the value of $\overline{m}^{(\mathrm{S})}$.

The implementation of this method used in this thesis is designed to be as automatic as possible and tries a number (typically 5–10) of different values for $\overline{m}^{(\mathrm{C})}$, including $\overline{m}^{(\mathrm{C})} = 1$ (no carving phase at all), to find the one giving the highest overall maximum log-posterior point in the second, SEB RPQ, phase of the process. Test results again suggest that the final outcomes are usually not sensitive to the exact values tried provided that they cover a reasonable part of the range 1–$\overline{m}^{(\mathrm{S})}$. The implementation is also designed to be efficient enough that a reasonably large number of different values for $\overline{m}^{(\mathrm{C})}$ can be tried without crippling the overall running time of the combined RPQ-MCMC process.

The heuristic method described here can be used for a large range of different values of c, the number of different initial states sought. The only limit is that there should be at least $\lceil \frac{c-1}{2} \rceil$ between states $S(\underline{k})$ and $S(k^\star)$ (including $S(\underline{k})$ but not including $S(k^\star)$), i.e., $\lceil \frac{c-1}{2} \rceil \leq k^\star - \underline{k}$. The implementation of the method used for this thesis adjusts even for this, reducing $\underline{k}$ if necessary so that $\lceil \frac{c-1}{2} \rceil = k^\star - \underline{k}$.

Some attention has to be paid to the possibility that a proper maximum log-posterior point will not be found in $\{S(k)\}$, i.e., that the log-posterior of the final state in the sequence is the highest of any state in the sequence and so a true maximum would presumably have been found if the RPQ had been allowed to run for longer. In the implementation used in this thesis the maximum is simply taken as the final state in the sequence generated by the RPQ. Provided that (as described above) $\overline{m}^{(\mathrm{S})}$ is reasonably large, the results of the testing of the implementation again suggest that it is not crucial to identify the true maximum log-posterior point.

Appendix G

Calculating $\widehat{R}_{\text{interval}}$

G.1 Introduction

This appendix describes the method used to calculate the interval-based $\widehat{R}$ diagnostic developed for this thesis to assess convergence of the Metropolis-Hastings SRP sampler described in Chapter 6. The method is based on the $\widehat{R}_{\text{interval}}$ diagnostic described in Brooks and Gelman (1998). A sequence of values for the interval-based $\widehat{R}$ is calculated over a moving sub-sequence of states that may become longer as the total sequence length increases. The method uses $c > 1$ uncoupled chains initiated at well-dispersed starting points.

G.2 $\widehat{R}_{\text{interval}}$

The sequence of SRP states for a chain with index i in a collection of c uncoupled chains in the MCMC process is denoted by $\{S_i(t)\}_{t\in\mathbb{Z}_+}$ where $S_i(t)$ is the state of the SRP immediately after transition t. The convergence diagnostic calculations use some scalar summary V of the full SRP state. Let $v_{i,j} = V(S_i(j = t))$, the scalar summary of the j^{th} state in the i^{th} chain. The sequence of scalar summaries for the i^{th} chain to transition t is $\{v_{i,j}\}_{j\in\{0,\ldots,t\}}$

For any one chain with index i in the collection of chains in the MCMC process, the sub-sequence of values used for the calculation of $\widehat{R}(t)$ after transition t is the last half of the total sequence of values $\{v_{i,j}\}_{j\in\{0,\ldots,t\}}$ for the chain. Thus after transition t the sub-sequence of values for chain i used for the calculation of $\widehat{R}(t)$ is $\{v_{i,j}\}_{j\in\{\lceil\frac{t}{2}\rceil,\ldots,t\}}$. The interval $I_{i,t}$ calculated for chain i after transition t is the empirical $100(1-\alpha)\%$ interval of $\{v_{i,j}\}_{j\in\{\lceil\frac{t}{2}\rceil,\ldots,t\}}$:

$$I_{i,t} = \left[100\frac{\alpha}{2} \text{ percentile of } \{v_{i,j}\}_{j\in\{\lceil\frac{t}{2}\rceil,\ldots,t\}},\ 100\left(1-\frac{\alpha}{2}\right) \text{ percentile of } \{v_{i,j}\}_{j\in\{\lceil\frac{t}{2}\rceil,\ldots,t\}}\right]$$

where α specifies the coverage of the interval. The implementation of the method used for the examples and results shown in this thesis uses $\alpha = 0.80$.

This calculation is repeated over each of the c chains in the process, giving c intervals $I_{1,t}, \ldots, I_{c,t}$ calculated after transition t. The entire set of values

$$\{v_{1,j}\}_{j\in\{\lceil\frac{t}{2}\rceil,\ldots,t\}} \cup \ldots \cup \{v_{c,j}\}_{j\in\{\lceil\frac{t}{2}\rceil,\ldots,t\}}$$

from all c sub-sequences is then used to calculate the total-chain interval $I_{.,t}$ at state t.

$\widehat{R}(t)$ is calculated as the ratio of the width of the total-chain interval after transition t to

the mean width of the within-chain intervals:

$$\widehat{R}(t) = \frac{\text{wid}\,(I_{.,t})}{\frac{1}{c}\sum_{i=1}^{c}\text{wid}\,(I_{i,t})}.$$

The calculation is relatively computationally expensive (interval widths cannot be efficiently updated as values are added to, and dropped from, the sub-sequences — a full recalculation of the interval and its width is required at each update) and the value of $\widehat{R}$ changes increasingly slowly as the total sequences become longer. It may be desirable to update the value of $\widehat{R}(t)$ only when $\delta(\nu)$ additional states have been added to the within-chain sub-sequences used for the calculation. $\delta(\nu)$ is referred to as the *sampling interval*.

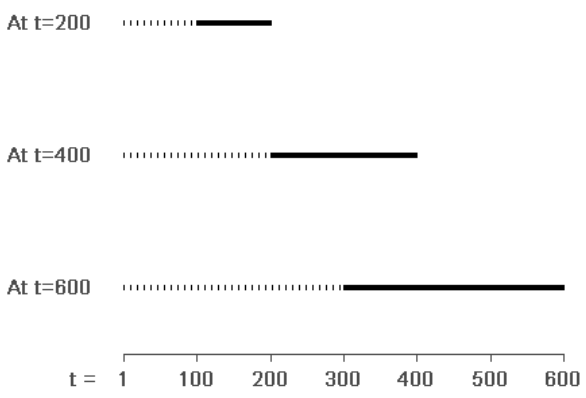


Figure G.1: Updating sub-sequences with $\delta(\nu) = 100$.

The implementation of the method used for the examples and results in this thesis uses a sampling interval $\delta(\nu) = 100$. Figure G.1 illustrates how this updates the sub-sequences used for calculating the within-chain interval widths when $\delta(\nu) = 100$. The solid lines indicate the states included in the calculations; the dotted lines indicate the states not included. The first calculation takes place after $t = 2\delta(\nu) = 200$. The interval for each chain is calculated using the last $\frac{t}{2} = 100$ values in the sequence of scalar values available for the chain. The calculation is updated at $t = 4\delta(\nu) = 400$. The interval for each chain is calculated using the last $\frac{t}{2} = 200$ values in the sequence of scalar values available for the chain. The next update is at $t = 6\delta(\nu) = 600$. The interval for each chain is calculated using the last $\frac{t}{2} = 300$ values in the sequence of scalar values available for the chain. As t increases the number of values used for the calculation of each interval increases and the overlap between the sub-sequences used at adjacent update points also increases.

If the data is multivariate and the number of data points n is large then the number of

transitions required before $\widehat{R}(t)$ is close to 1 can be very large (order 10^6 or more). The number of scalar values used in the calculation of each interval also becomes very large, even only using the second half of the total sequence of scalar values. This not only causes the calculation of the interval widths to become increasingly time-consuming, but the very large overlaps between the values used for each successive interval calculation also mean that the values of $\widehat{R}(t)$ calculated change only very slowly. To mitigate these problems it may also be necessary to specify a maximum number $\overline{\nu}$ of scalar summary values to be used for the calculation of each within-chain interval. The examples and results shown in this dissertation use $\overline{\nu} = 1,000,000$ unless stated otherwise.

In the implementation used for this thesis (with $\delta(\nu) = 100$) the value of $\widehat{R}(1)$ is set to 0, and is not updated until $t = 2\delta(\nu) = 200$ as described above. The values of $\widehat{R}(201), \ldots \widehat{R}(399)$ are all the same as $\widehat{R}(200)$, until the updated value is calculated at $t = 4\delta(\nu) = 400$, etc. Once $t - \lceil \frac{t}{2} \rceil + 1 \geq \overline{\nu}$ the updates are calculated every time the total number of transitions increases by $\delta(\nu)$.

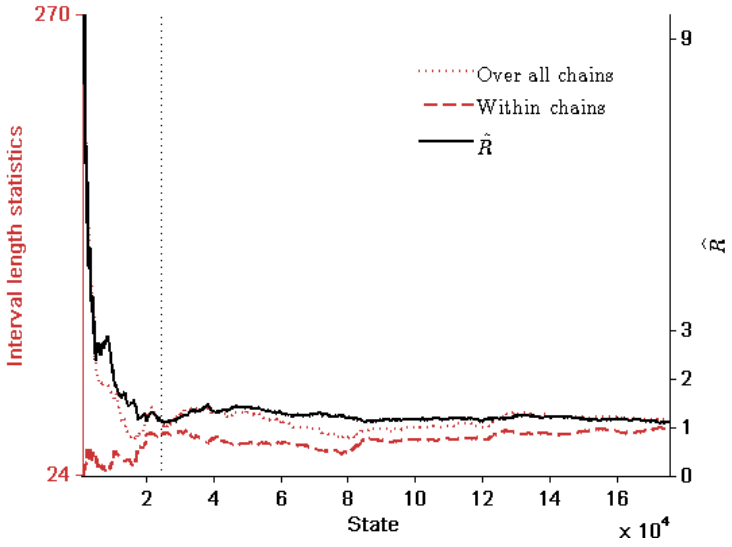


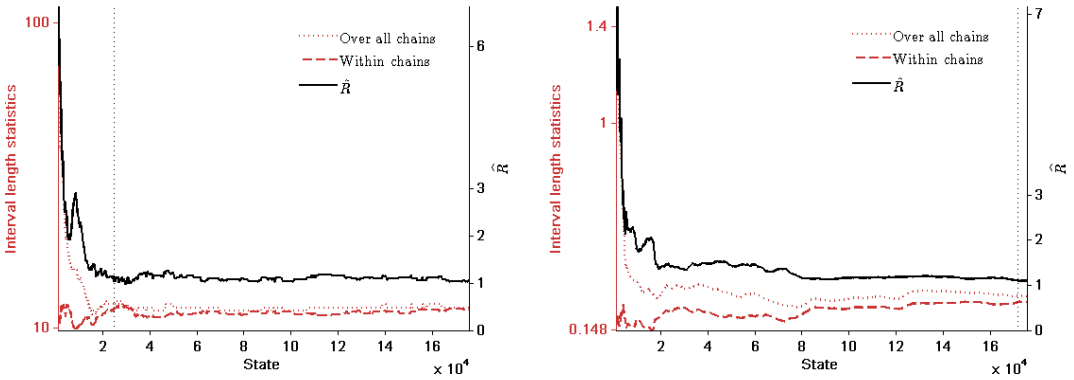
Figure G.2: $\widehat{R}$ calculations with V = number of leaves.

G.3 Trace plots

Figure G.2 shows plots of the sequence $\{\widehat{R}(t)\}$ thus calculated together with the components of the calculation, the mean within-chain interval width and the interval width over all chains. These traces relate to an SRP MCMC process to make a density estimate using a sample of $n = 50,000$ data points drawn from example Density I, $d = 2$ (see Appendix B). The scalar summary of SRP state used is the number of leaves in the SRP. The scale for the interval widths is on the left-hand vertical axis, the scale for $\{\widehat{R}(t)\}$ is on the right-hand vertical

axis. The horizontal axis only shows states from $t = 1,000$ until convergence is eventually diagnosed (using number of leaves, number of cherries, and average leaf depth scalar summaries together). The early values of $\{\widehat{R}(t)\}$ are typically very large if the over-diversified initial states for each chain described in Appendix F are used but fall rapidly, as in Figure G.2. However, as Figure G.2 also shows, the sequence $\{\widehat{R}(t)\}$ does not move monotonically. The uneven pattern of values shown here is typical of the traces examined during the testing carried out for this thesis for many different densities and dimensions. There are quite complex interactions between the respective effects on the mean within-chain interval width and the total-chain interval width of adding states to, and dropping states from, each within-chain sub-sequence.

Figure G.3(a) shows the equivalent of Figure G.2 for the same MCMC process using the number of cherries as the scalar summary of SRP state, while Figure G.3(b) uses the average leaf depth scalar summary.



(a) V = number of cherries.

(b) V = average leaf depth.

Figure G.3: Calculating $\widehat{R}$ with different scalars summaries.

Different methods could be used to create the sub-sequences of scalar values used each time that $\widehat{R}(t)$ is updated, including using non-overlapping sub-sequences similar to the batches described in Brooks and Gelman (1998). The approach described here and implemented for this thesis seems to provide a reasonable compromise between using a large number of values in each sub-sequence (which reduces the short-term variability of the values of $\widehat{R}(t)$ calculated) and gradually dropping the scalar summary values calculated from the earlier SRP states in the chains. Other values for the sampling interval $\delta(\nu)$ could also be used. $\delta(\nu) = 1$ would update the value of $\{\widehat{R}(t)\}$ at every iteration of the MCMC process and would add to the computational burden of the convergence calculations while only bringing forward the diagnosis of convergence by a maximum of $\delta(\nu) - 1$ iterations. The sampling interval $\delta(\nu) = 100$ used in the implementation of the method for most of the examples shown in this thesis is chosen as a compromise between the speed gained by omitting some diagnostic calculations and the potential waste of a large number of iterations of the process if a large

sampling interval delays the diagnosis of convergence and hence the start of sampling from the chains.

Whatever method is used to create the sub-sequences of scalar values used to calculate $\widehat{R}(t)$, it is important that the scalar summaries from the early states in the chains, the states most influenced by the initial states, are excluded from the calculations. Gelman and Rubin (1992) and Brooks and Gelman (1998) discuss this and suggest that including the early states can lead to an unnecessarily delayed diagnosis of convergence because of the high variability of the early states. The results of testing various convergence diagnostics during the research for this thesis, however, suggests that (at least for the MCMC process used here) including the early states can in fact give a false *early* diagnosis of convergence. This is seems to be because including the widely dispersed initial states can inflate the within-chain variability measure more than the between-chains variability measure and give a lower $\widehat{R}$ earlier than would be the case if the initial states were to be excluded.

Figure G.4 illustrates this phenomenon, showing the sequence of interval-based $\widehat{R}$s, and the values of the mean within-chain interval width and the total-chain interval width used to calculate each $\widehat{R}$, for the same sample data as for Figure G.2 and calculated as described above except that number of leaves scalar summaries of *all* states in each chain are included in the sub-sequences used for the intervals. Figure G.4 is shown using the same horizontal scale as Figure G.2 to facilitate the comparison. In Figure G.2 the convergence is diagnosed only when $\widehat{R}$s using all three scalar summaries (number of leaves, number of cherries, average leaf depth) are considered to be close enough to 1. The same criterion applies in Figure G.4, but the values of $\widehat{R}$ for all of these scalars fall misleadingly quickly when the initial states are included in the calculations.[1]

Brooks and Gelman (1998) point out that convergence should not just be assessed by looking for values of $\widehat{R}$ close enough to 1. Traces of the actual variability measures (for example, the estimates of within and between chain variance, for $\widehat{R}_{\text{variance}}$, or the mean within-chain interval width and total-chain interval width) must be considered as well. Convergence cannot be said to have taken place until the sequences of these values have also stabilised. Figure G.4 shows clearly that both the mean within-chain interval width and total-chain interval width sequences have not stabilised when the value of $\widehat{R}$ alone, calculated using all states in the chains, falsely suggests that convergence has taken place.

It is also useful to consider traces of $\widehat{R}$ sequences for different scalar summaries. During the testing carried out for this thesis it was very common to find that one or two of the three scalar summaries used here (number of leaves, number of cherries, average leaf depth) might suggest that convergence had been achieved much later than the other(s). Using more than one scalar summary is especially important if some form of automated assessment of convergence is used (whereby the sampling method looks for a low $\widehat{R}$ value to trigger the start of sampling).

[1] The same effect was observed in testing a $\widehat{R}_{\text{variance}}$ diagnostic, but this was not ultimately used for the results and examples included in this thesis.

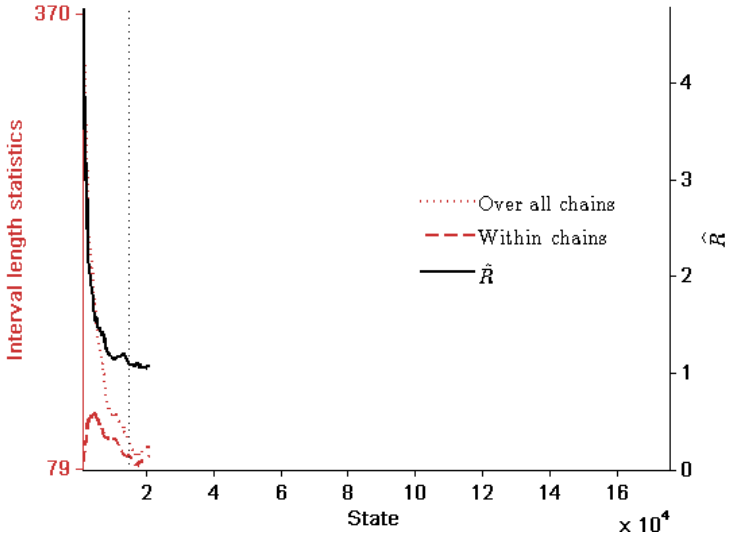


Figure G.4: $\widehat{R}$ calculations with V = number of leaves, using all states.

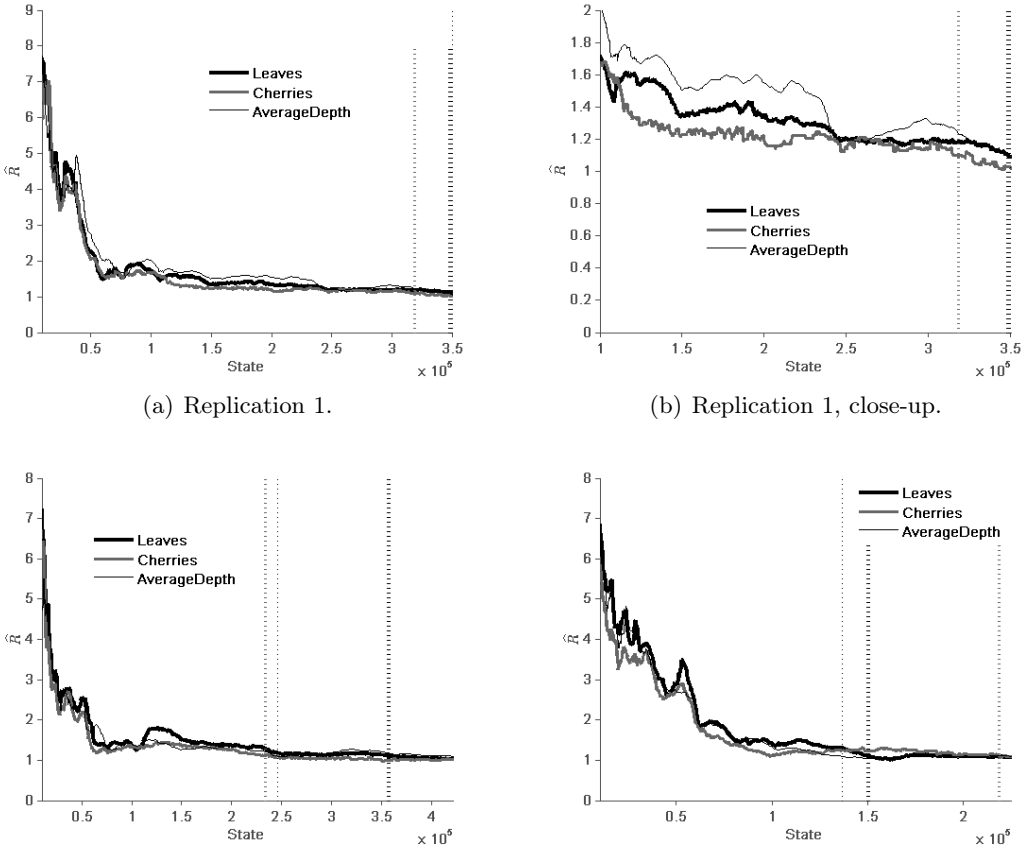
Testing the MCMC process with $\widehat{R}$ sequences for all three scalar summaries using repeated replications (using different pseudo-random number sequences and data samples) shows that the scalar summary which first, or last, gives $\widehat{R}$ close to 1 (and the number of transitions required for this to take place) can vary between replications. This indicates that none of the scalar summaries used here is generally most or least sensitive to the convergence of the chains, nor is any of these scalars redundant in the sense of giving no further useful information in addition to that available from the other two.

Figure G.5 illustrates this using three replications of an MCMC process to create an averaged RMRP histogram density estimate using sample data drawn from a 3-d multivariate standard Gaussian distribution (example Density III in Appendix B). In each sub-figure the three vertical dotted lines show the transitions at which convergence is diagnosed using the $\widehat{R}$ convergence diagnostic calculated using each of the three scalar summaries.

Figure G.5 also shows a very typical feature of a sequence of $\widehat{R}$ values calculated using the gradually lengthening and moving sub-sequences of SRP states described above. A relatively early and substantial fall in the $\widehat{R}$ followed by a subsequent substantial rise is often observed. The early fall in $\widehat{R}$ occurs as the states most influenced by the intentionally well-dispersed starting states are dropped from the sub-sequences used in the calculation, but there is still considerable variability within each chain and this is reflected in the subsequent rise in $\widehat{R}$. Requiring the $\widehat{R}$ values using all three scalar summaries to all be close to 1 can help to prevent diagnosing convergence too early, especially if automated methods are using to assess convergence and trigger sampling.

Finally, the implementation of the convergence heuristic used in this thesis allows a maximum chain length (number of transitions) to be specified so that the process will be halted if

convergence is not diagnosed before this maximum is reached.



(a) Replication 1.

(b) Replication 1, close-up.

(c) Replication 2.

(d) Replication 3.

Figure G.5: $\widehat{R}$ from three replications to estimate a 3-d Gaussian density.

Appendix H

Sampling from an SRP MCMC process

H.1 Introduction

This appendix discusses how to sample from the chains in an SRP MCMC process once convergence has been achieved. In the implementation used in this thesis samples are taken from all c chains used to diagnose convergence. Unless otherwise stated, all results in this Appendix are based on samples taken across three chains.

H.2 Choosing the number of samples and the thin-out value

The computational cost of the sampling stage of the MCMC process has two components: the cost of making the total number of transitions required, and the cost of averaging the samples taken. In the first implementation of the SRP MCMC process, developed for Sainudiin et al. (2013), the number of samples averaged was effectively restricted by memory requirements and the per-state-sampled cost was proportional to the size n of the data set ${}^{n}X$ associated with the SRP. Thinning was usually necessary to be able to carry out the averaging process. However, after improvements in the implementation, the only memory restriction now applies to the complexity of the average itself, not the number of SRP samples on which it is based, and the per-sample cost is related to n only to the extent that the complexity of the SRPs being averaged increases with n (see Section 6.5.3).

The total number of transitions required to obtain the samples once convergence as been achieved is referred to as the number of *sampling transitions*). Sampling 10,000 states from a single chain with a thin-out of $r = 1$ requires 10,000 sampling transitions. Sampling 100 states with a thin-out of 100 also requires 10,000 sampling transitions. Sampling 10,000 states with a thin-out of 100 requires 1,000,000 sampling transitions. The cost of each transition increases with both the data sample size n and the number of dimensions in the data.

If the number of sampling transitions is too short then the SRP sample may only cover a small part of the support of the posterior even if the chain has reached the target distribution.

The number of leaves in the final average will increase with the total number of samples taken: the larger the number of SRP samples used, the larger the number of leaf nodes in the final average. In addition the total number of transitions over which sampling takes place also influences the number of leaves in the final average, and in fact can have more affect on this than the number of samples taken.

Table H.1 demonstrates this. The table is organised to group together the results for a specified total number of sampling transitions. Within each group the average number of leaves m, over 10 replications of the process, in the final average RMRPs using thin-out values

$r = 1$, 10, 100, and 1,000 is shown. The average time taken for the whole process (including finding the initial states and burning in the chains) is also shown. The number of samples taken (total number of transitions divided by r) is also indicated. These results are obtained using the MCMC process described in this thesis with three chains and a data sample of 50,000 data points drawn from example Density III (see Appendix B), the standard Gaussian density, with $d = 1$.

The results in Table H.1 suggest that taking the same number of SRP samples over a longer length of chain (i.e., using a higher thin-out value) can increase the number of leaves in the final density estimate more than taking a larger number of SRP samples over the same number of transitions (lowering the thin-out value). In addition, most of the cost of the sampling phase relates to averaging the samples, so taking the same number of SRP samples from a longer length of chain is also a computationally efficient way to increase the number of leaves in the final RMRP density estimate.

However, approximations of the L_1 errors for the density estimates against the true density (using quasi-Monte Carlo integration over $1,000,000$ quasi-random points, as in Chapter 7) showed that obtaining an estimate with more leaves by spreading the samples over a longer length of chain does not necessarily increase the accuracy of the estimate. The average approximate L_1 error for the final average RMRP density estimates for each combination of N and r in Table H.1 was close to 0.034 except for the single sample $N = 1$ case when the average estimated error was slightly higher (0.038).

The same tests repeated for example Density III, the standard Gaussian density, with $d = 3$ are shown in Table H.2 (results over 10^7 sampling transitions are not available due to lack of time). Spreading the samples over more total transitions again resulted in a final average RMRP with more leaves, but the average approximate L_1 error for each combination of N and r was close to 0.23 except for the single sample $N = 1$ case (0.24).

This is due to the interaction of the natural Catalan prior and the size of the sample data (see Section 6.5.3). The support of the posterior is strongly influenced by this prior, under which the chains will explore more deeply split states when the sample size is increased. For a fixed sample size the posterior distribution is concentrated on a small number of states (relative to the total state space). The averaging operation results in an RMRP with a partition that is the union of the partitions of the individual RMRP histograms averaged (see Algorithm 3.6, Section 3.5), so that each minor difference in the partitions of the RMRPs averaged contributes to the complexity of the result, but with relatively little effect on the overall 'shape' of the estimate because almost all of the states averaged will come from a relatively small and concentrated subset of the state space.

Table H.3 shows the effect of increasing the data sample size to $n = 100,000$ on the average number of leaves in the final density estimate (and the time taken to make it). The results in Table H.3 were obtained by repeating a subset of the tests used for Table H.2 with this

Table H.1: The influence of total number of sampling transitions and number of SRP samples on the leaves in the averaged RMRP, Density III ($d = 1$), $n = 50,000$.

Thin-out r	1	10	100	1000
Total transitions 1				
N	1			
Av. leaves	65			
Av. time (s)	4			
Total transitions 10^2				
N	10^2	10		
Av. leaves	103	100		
Av. time (s)	4	5		
Total transitions 10^3				
N	10^3	10^2	10	
Av. leaves	160	156	125	
Av. time (s)	6	6	6	
Total transitions 10^4				
N	10^4	10^3	10^2	10
Av. leaves	325	317	259	140
Av. time (s)	13	7	6	6
Total transitions 10^5				
N	10^5	10^4	10^3	10^2
Av. leaves	721	703	582	318
Av. time (s)	170	24	10	9
Total transitions 10^6				
N	10^6	10^5	10^4	10^3
Av. leaves	1601	1563	1297	728
Av. time (s)	4602	458	53	20
Total transitions 1×10^7				
N	10^7	10^6	10^5	10^4
Av. leaves	3491	3412	2851	1617
Av. time (s)	72996	6642	562	94

larger data sample. The approximated L_1 errors for these density estimates were all around 0.20, in contrast to 0.23 for density estimates made for Table H.2. The only effective way to increase the accuracy at of the density estimate, using the present implementation of the MCMC process with the natural Catalan prior, is to increase the sample size. As Table H.3 shows, the computational cost of obtaining the MCMC density estimate is then considerably higher.

Tests were carried out using data drawn from Example Density II, $d = 1$, to investigate the effect of the number of sampling transitions and data sample size further. Convergence was diagnosed using three chains and the $\widehat{R}$ method described in Section 6.6 but samples were taken from only one of the post-burn-in chains. These samples were used to form two RMRP density estimates, one averaging the SRP samples collected from the first half of the chain after

Table H.2: The influence of total number of sampling transitions and number of SRP samples on the leaves in the averaged RMRP, Density III ($d = 3$), $n = 50,000$.

Thin-out r	1	10	100	1000
Total transitions 1				
N	1			
Av. leaves	1159			
Av. time (s)	384			
Total transitions 10^2				
N	10^2	10		
Av. leaves	1657	1655		
Av. time (s)	326	327		
Total transitions 10^3				
N	10^3	10^2	10	
Av. leaves	1767	1763	1727	
Av. time (s)	403	318	305	
Total transitions 10^4				
N	10^4	10^3	10^2	10
Av. leaves	2458	2453	2409	2069
Av. time (s)	402	335	311	348
Total transitions 10^5				
N	10^5	10^4	10^3	10^2
Av. leaves	4642	4635	4567	4020
Av. time (s)	1371	415	294	294
Total transitions 10^6				
N	10^6	10^5	10^4	10^3
Av. leaves	9511	9578	9450	8412
Av. time (s)	19341	2510	695	537

Table H.3: The influence of total number of sampling transitions and number of SRP samples on the leaves in the averaged RMRP, Density III ($d = 3$), $n = 100,000$.

Thin-out r	100	1000
Total transitions 10^4		
N	10^2	10
Av. leaves	3399	3037
Av. time (s)	2046	2297
Total transitions 10^5		
N	10^3	10^2
Av. leaves	6231	5689
Av. time (s)	1672	1738
Total transitions 10^6		
N	10^4	10^3
Av. leaves	13067	12024
Av. time (s)	3066	2269

burn-in, the other averaging samples collected from the second half of the chain after burn-in. The approximate L_1 error was calculated for both these average RMRPs. For comparison, the approximate L_1 error was also calculated for the first SRP histogram in each of the lengths of chain over which the samples for each average was taken, giving L_1 errors for two individual histograms in each half of the post-burn-in chain.

These tests were repeated over 10 replications of the process (with different data samples and a different sequence of pseudo-random numbers used in the MCMC process) for thin-out $r = 100$, total number of samples in each separate half of the post-burn-in chain $N = 1,000$, and number of data points in the data sample $n = 50,000$. The entire process was then repeated with $N = 10,000$ (i.e., sampling more from a longer length of chain), and then again for $n = 100,000$ (i.e., a large data sample). The results are summarised in Table H.4. The first SRP histogram state in the first half of the post-burn-in chain is denoted by $S^{(1)}(1)$ and the first SRP histogram state in the second half of the post-burn-in chain is denoted by $S^{(2)}(1)$. The RMRP density estimate from the first half of the post-burn-in chain is denoted by ${}^{\square}\bar{f}^{(1)}$ and the average RMRP density estimate from the second half of the post-burn-in chain is denoted by ${}^{\square}\bar{f}^{(2)}$. Approximate L_1 errors ($\hat{L}_1$) are given to 3 decimal places.

Table H.4: The influence of thin-out r, number of SRP samples N, and size of data sample n on approximated L_1 error and number of leaves, Density II ($d = 1$).

		$S^{(1)}(1)$	$S^{(2)}(1)$	${}^{\square}\bar{f}^{(1)}$	${}^{\square}\bar{f}^{(2)}$
$r = 100$, $N = 1,000$, $n = 50,000$					
$\hat{L}_1$	Average	0.042	0.042	0.037	0.037
	Range	0.039–0.046	0.038–0.045	0.033–0.040	0.033–0.039
Leaves	Average	70	69	6005	591
	Range	58–79	57–88	501–746	489–649
$r = 100$, $N = 10,000$, $n = 50,000$					
$\hat{L}_1$	Average	0.042	0.041	0.037	0.037
	Range	0.039–0.046	0.037–0.047	0.033–0.040	0.033–0.040
Leaves	Average	70	76	1374	1348
	Range	58–79	64–101	1144–1645	1157–1567
$r = 100$, $N = 1,000$, $n = 100,000$					
$\hat{L}_1$	Average	0.034	0.035	0.031	0.031
	Range	0.032–0.039	0.032–0.038	0.028–0.036	0.027–0.036
Leaves	Average	82	83	689	702
	Range	70–99	71–99	630–752	626–759

Tests (results not shown) on other individual SRP histogram states sampled from these chains gave a similar range and mean for the approximate L_1 as for the single SRP states $S^{(1)}(1)$ and $S^{(2)}(1)$ in Table H.4.

Repeating these tests with $d = 3$ gave the results shown in Table H.5.

Tests (results not shown) on other individual SRP histogram states sampled from these

Table H.5: The influence of thin-out r, number of SRP samples N, and size of data sample n on approximated L_1 error and number of leaves, Density II ($d = 1$).

		$S^{(1)}(1)$	$S^{(2)}(1)$	$\square\bar{f}^{(1)}$	$\square\bar{f}^{(2)}$
$r = 100$, $N = 1,000$, $n = 50,000$					
$\hat{L}_1$	Average	0.319	0.319	0.300	0.300
	Range	0.312–0.327	0.294–0.326	0.294–0.308	0.295–0.307
Leaves	Average	1998	1984	6240	6268
	Range	1842–2091	1855–2035	5926–6425	5905–6486
$r = 100$, $N = 10,000$, $n = 50,000$					
$\hat{L}_1$	Average	0.319	0.318	0.295	0.296
	Range	0.313–0.327	0.290–0.327	0.291–0.303	0.290–0.304
Leaves	Average	1998	2035	12783	12816
	Range	1842–2054	1917–2146	13147–12350	12345–13106
$r = 100$, $N = 1,000$, $n = 100,000$					
$\hat{L}_1$	Average	0.277	0.277	0.261	0.260
	Range	0.273–0.282	0.272–0.281	0.257–0.265	0.257–0.265
Leaves	Average	3131	3085	8571	8552
	Range	2972–3279	2910–3180	8259–8729	8240–8733

chains again gave a similar range and mean for the approximate L_1 as for the single SRP states $S^{(1)}(1)$ and $S^{(2)}(1)$ in Table H.5.

A final result from the tests described in this Appendix is that there is no evidence that the approximate L_1 error *increases* if the number of leaves in the final average RMRP is increased by spreading the SRP samples over more sampling transitions. This is important because, taken together with the results showing that the error reduces as the data sample size n increases, it gives some empirical evidence that the sample mean estimate RMRP may be regarded as an L_1-*consistent* density estimate, i.e., that (no matter what effect the thin-out value and number of SRP states sampled has on the complexity of the average RMRP estimate) the L_1 error falls as the size of the data sample increases.

The times given in this appendix were recorded when other processes were running on the same machine.

Appendix I

An independent Metropolis-Hastings sampler for SRPs

I.1 Introduction

This appendix discusses some preliminary research into the possibility of developing an independent Metropolis-Hastings sampler for SRP states. Using an independent Metropolis-Hastings sampler with proposal distribution g the probability $g(\,^{\circ}s')$ of a proposal $\,^{\circ}s'$ is independent of the current state $\,^{\circ}s$ and the acceptance probability is

$$a(\,^{\circ}s,\,^{\circ}s') := \min\left\{1, \frac{\pi(\,^{\circ}s')g(\,^{\circ}s)}{\pi(\,^{\circ}s)g(\,^{\circ}s')}\right\} \quad .$$

Ideally the proposal distribution g should be close to the target distribution f, and it is important that g should dominate f in the tails (Bolstad 2010, chap. 6).

The form of independent Metropolis-Hastings sampler considered here uses proposal distributions of the form $g(\,^{\circ}s') = g_{k'}(\,^{\circ}s', |\,^{\circ}s' \in {}^{\circ}\mathbb{S}_{k'}) \cdot g_{\text{splits}}(K = k')$ where $g_{k'}(\,^{\circ}s' \,|\, ^{\circ}s' \in {}^{\circ}\mathbb{S}_{k'})$ is some probability mass distribution on the $C_{k'}$ unique SRP states with k' splits and K is a random number of splits with probability mass function $g_{\text{splits}}(K = k')$ and a finite state space $K \in \{0, 1, \ldots, \overline{m} - 1\}$ defined by some maximum number of leaves $\overline{m} < \infty$.

I.2 Binary tree probabilities

This section discusses two possible forms of $g_{k'}(\,^{\circ}s' \,|\, ^{\circ}s' \in {}^{\circ}\mathbb{S}_{k'})$, both of which exploit the binary tree structure of SRPs.

I.2.1 The natural distribution of binary tree states

Appendix C shows how C_k, the number of unique binary trees with $m = |\mathbb{L}(s)| = k + 1$ leaves (k splits), increases with k and briefly discusses how algorithms that progressively grow a binary tree by selecting uniformly at random a leaf node to be the next to be split can be analysed by considering the number of different routes from the root node to each binary tree state in the state space. This kind of growth process is a very 'natural' one. In this appendix the probability of reaching a particular state $\,^{\circ}s \in {}^{\circ}\mathbb{S}_{k'}$ after exactly k' splits using this natural growth process is referred to as the *natural probability* and the probability mass distribution on ${}^{\circ}\mathbb{S}_{k'}$ that assigns each state in $\,^{\circ}s \in {}^{\circ}\mathbb{S}_{k'}$ its natural probability is referred to as the *natural probability distribution* of states in ${}^{\circ}\mathbb{S}_{k'}$.

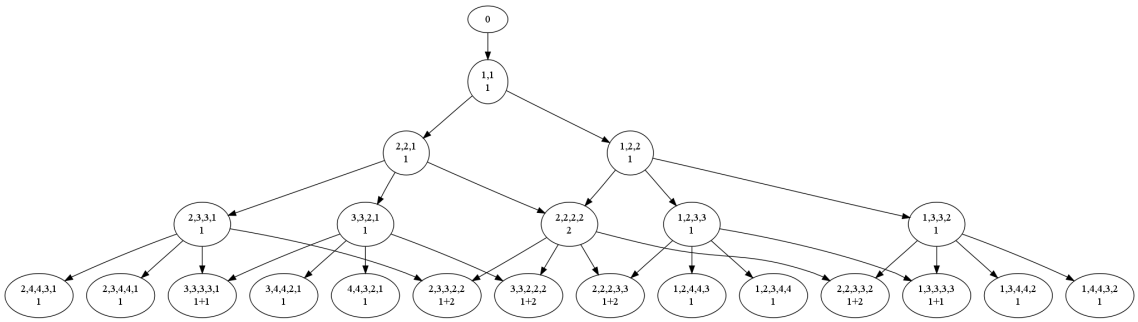


Figure I.1: State space for $m = 5$ leaves ($k = 4$).

For example, consider the five trees with exactly $m = |\mathbb{L}(s)| = 4$ leaves (the states in $^{\circ}\mathbb{S}_3$) in Figure I.1 (Figure I.1 is identical to Figure C.1). The natural probability of the tree with state labelled '2,2,2,2' is $\frac{2}{6}$ and the other four states in $^{\circ}\mathbb{S}_3$ each have natural probability $\frac{1}{6}$. Now consider the 14 trees with exactly $m = |\mathbb{L}(s)| = 5$ leaves (the states in $^{\circ}\mathbb{S}_4$) in Figure I.1. The total number of routes to these states is 4!=24 (see Appendix C). The natural probability of the tree with state labelled '2,4,4,3,1' is $\frac{1}{24}$. The natural probability of the tree with state labelled '3,3,3,3,1' is $\frac{2}{24}$. The natural probability of the tree with state labelled '2,3,3,2,2' is $\frac{3}{24}$. The more routes through the state space that there are from the root node to a unique tree state, the higher the natural probability of that tree state.

As k' increases the probability mass is increasingly concentrated on a smaller proportion of the total $C_{k'}$ possible states in $^{\circ}\mathbb{S}_{k'}$ and the proportion of relatively low probability states gets larger and larger. This suggests that the natural probability distribution is not a good component of the proposal distribution g for an independent Metropolis-Hastings sampler.

I.2.2 A uniform distribution for binary tree states

An obvious alternative to the natural distribution of binary tree states is a uniform distribution

$$g_{k'}(\,^{\circ}s' \mid \,^{\circ}s' \in \,^{\circ}\mathbb{S}_{k'}) = \frac{1}{C_{k'}} \quad .$$

Although $C_{k'}$ itself may be very large it is possible to draw an SRP $^{\circ}s'$ uniformly at random from $^{\circ}\mathbb{S}_{k'}$, the space of all SRPs with k' splits, using an adaptation of Algorithm W (*Uniformly random strings of nested parentheses*) given in Knuth (2006a).

I.3 The prior and the acceptance probability

A uniform proposal distribution g on the finite state space $^{\circ}\mathbb{S}_{0:\overline{m}-1}$ of SRPs with a maximum of $\overline{m}$ leaves can be created using $g_{k'}(\,^{\circ}s' \mid \,^{\circ}s' \in \,^{\circ}\mathbb{S}_{k'}) = \frac{1}{C_{k'}}$ and $g_{\text{splits}}(K = k') = \frac{1}{\overline{m}}$,

$k' = 0, 1, \ldots, \overline{m} - 1$. The acceptance probability is

$$a(\,^\circ s,\, ^\circ s') := \min\left\{1, \frac{\pi(\,^\circ s')g(\,^\circ s)}{\pi(\,^\circ s)g(\,^\circ s')}\right\}$$

and

$$\begin{aligned}
\frac{\pi(\,^\circ s')g(\,^\circ s)}{\pi(\,^\circ s)g(\,^\circ s')} &= \frac{\mathcal{L}\left(\hat{f}'_n\right)}{\mathcal{L}\left(\hat{f}_n\right)} \frac{\Pr\{\,^\circ s'\}}{\Pr\{\,^\circ s\}} \frac{g(\,^\circ s)}{g(\,^\circ s')} \\
&= \frac{\mathcal{L}\left(\hat{f}'_n\right)}{\mathcal{L}\left(\hat{f}_n\right)} \frac{\Pr\{\,^\circ s'\}}{\Pr\{\,^\circ s\}} \frac{g_k(\,^\circ s \mid \,^\circ s \in \,^\circ\mathbb{S}_k)}{g_{k'}(\,^\circ s' \mid \,^\circ s' \in \,^\circ\mathbb{S}_{k'})} \frac{g_{\text{splits}}(k)}{g_{\text{splits}}(k')} \\
&= \frac{\mathcal{L}\left(\hat{f}'_n\right)}{\mathcal{L}\left(\hat{f}_n\right)} \frac{\Pr\{\,^\circ s'\}}{\Pr\{\,^\circ s\}} \frac{C_{k'}}{C_k} \quad ,
\end{aligned}$$

where $\hat{f}'_n$ and $\hat{f}_n$ are the histograms based on the partitions of $^\circ s'$ and $^\circ s$ respectively and $^\circ s \in \,^\circ\mathbb{S}_k$.

If the natural Catalan prior $\Pr\{\,^\circ s\} = \sum_{k=0}^{\infty} \mathbb{1}_{\,^\circ\mathbb{S}_k}(\,^\circ s) \frac{1}{aC_k^2}$ (Equation (6.2)) is used then

$$\begin{aligned}
\frac{\pi(\,^\circ s')g(\,^\circ s')}{\pi(\,^\circ s)g(\,^\circ s)} &= \frac{\mathcal{L}\left(\hat{f}'_n\right)}{\mathcal{L}\left(\hat{f}_n\right)} \frac{\Pr\{\,^\circ s'\}}{\Pr\{\,^\circ s\}} \frac{C_{k'}}{C_k} \\
&= \frac{\mathcal{L}\left(\hat{f}'_n\right)}{\mathcal{L}\left(\hat{f}_n\right)} \left(\frac{C_k}{C_{k'}}\right)^2 \frac{C_{k'}}{C_k} \\
&= \frac{\mathcal{L}\left(\hat{f}'_n\right)}{\mathcal{L}\left(\hat{f}_n\right)} \frac{C_k}{C_{k'}} \quad .
\end{aligned}$$

As $k \to \infty$, $\frac{C_k}{C_{k+1}} = \frac{(k+2)}{2(2k+1)} \to \frac{1}{4} = \left(\frac{1}{2}\right)^2$ from above and so when $|k' - k|$ is reasonably large,

$$\frac{C_k}{C_{k'}} \approx \left(\frac{1}{2}\right)^{2(k'-k)} = 2^{2(k-k')} \quad .$$

A computerised implementation of an independent Metropolis-Hastings sampler such as the one described here also has to deal with the possibility that unavoidable restrictions on the state space (Appendix C) may mean that $^\circ\mathbb{S}_{0:\overline{m}-1} \not\subseteq \,^\circ\tilde{\mathbb{S}}$ where $^\circ\tilde{\mathbb{S}}$ is the largest possible state space including states with $\overline{m}$ leaves permitted by the computer implementation.

I.4 Mixing

In theory an independent Metropolis-Hastings sampler may give better mixing than the sampler using the stay-split-merge base chain described in Chapter 6.1. However, the overall acceptance rate for each transition under the stay-split-merge base is reasonably high whereas the analysis of the acceptance probability above suggests that the rate of acceptance of proposals using this independent Metropolis-Hastings sampler, with the natural Catalan prior, may be quite low especially if a large maximum number of leaves $\overline{m}$ is permitted. The state space, although finite, is very large indeed and most of these states will have very low SRP histogram likelihood (the posterior is concentrated on a small subset of states). This has indeed been the case in the very limited experimentation with the independent Metropolis Hastings sampler described here carried out during the course of this thesis.

Appendix J

Evaluating RMRP approximations to a kernel density estimate

J.1 Estimated errors in estimating Density II using a KDE and an RMRP approximation to the KDE

Tables J.1, J.2, J.3, and J.4 show the full results discussed in Section 7.5. The tables show estimates for $\hat{d}_{KL}$ and $\hat{L}_1$ for true multivariate density Density II and the estimate $\hat{f}$ as, first, the KDE (with $n_K = 2,000$), and then the RMRP approximation to the KDE with $\overline{\psi}$ = 0.00005–0.0000001 for $d = 2, 3, 4$ and 5. The Kullback-Leibler loss ($\hat{d}_{KL}$) was estimated using Equation (7.2) and $N = 1,000,000$. The L_1 error ($\hat{L}_1$ error) was estimated using quasi-Monte Carlo integration (Niederreiter 1992) over $N = 1,000,000$ quasi-random points and Equation (7.3).

The time taken for each estimate and the numbers of leaves in the RMRP approximations are also shown. Estimated errors are given to two decimal places, KDE times are rounded to indicate the general magnitude of times from several different replications, and the RMRP approximation times are given to one decimal place. Other than the KDE times, these result shown are from just one replication of the process.

The values of $\hat{d}_{KL}$ and $\hat{L}_1$ and the number of leaves in the RMRP did not vary greatly over a limited number of repetitions with different data sets and different sequences of pseudo-random numbers in the RPQ process (results not shown). The timings were recorded when other processes running on the same machine and can only be taken as a general indication of the time required for each process.

Table J.1: 2-d case: estimated errors for KDE and RMRP-KDE approximations.

	$\hat{d}_{KL}$	$\hat{L}_1$ error	Time (s)	Leaves
KDE ($n_K = 2,000$)	0.04	0.20	5,000–7,200	n/a
RMRP-KDE approximations				
$\overline{\psi} = 0.0001$	1.13	0.24	0.4	941
$\overline{\psi} = 0.00005$	0.32	0.22	0.8	1,552
$\overline{\psi} = 0.00001$	0.15	0.21	2.1	4,501
$\overline{\psi} = 0.000005$	0.15	0.20	3.8	7,322
$\overline{\psi} = 0.000001$	0.15	0.20	10.2	21,105
$\overline{\psi} = 0.0000005$	0.15	0.20	17.7	33,285
$\overline{\psi} = 0.0000001$	0.15	0.20	47.5	97,721

Table J.2: 3-d case: estimated errors for KDE and RMRP-KDE approximations.

	$\hat{d}_{KL}$	$\hat{L}_1$ error	Time (s)	Leaves
KDE ($n_K = 2,000$)	0.13	0.35	5,600–7,200	n/a
RMRP-KDE approximations				
$\overline{\psi} = 0.0001$	1.77	0.44	0.7	1,675
$\overline{\psi} = 0.00005$	1.76	0.43	1.3	2,804
$\overline{\psi} = 0.00001$	1.74	0.41	4.2	8,923
$\overline{\psi} = 0.000005$	1.74	0.41	7.7	14,906
$\overline{\psi} = 0.000001$	1.72	0.40	23.7	49,480
$\overline{\psi} = 0.0000005$	1.72	0.40	44.1	83,488
$\overline{\psi} = 0.0000001$	1.72	0.40	132.7	274,633

Table J.3: 4-d case: estimated errors for KDE and RMRP-KDE approximations.

	$\hat{d}_{KL}$	$\hat{L}_1$ error	Time (s)	Leaves
KDE ($n_K = 2,000$)	0.25	0.51	7,200–8,050	n/a
RMRP-KDE approximations				
$\overline{\psi} = 0.0001$	4.43	0.70	1.7	2,227
$\overline{\psi} = 0.00005$	4.22	0.65	3.6	4,092
$\overline{\psi} = 0.00001$	4.17	0.62	12.6	14,876
$\overline{\psi} = 0.000005$	4.13	0.61	23.8	25,925
$\overline{\psi} = 0.000001$	4.11	0.59	78.3	90,621
$\overline{\psi} = 0.0000005$	4.11	0.59	149.7	158,181
$\overline{\psi} = 0.0000001$	4.10	0.58	484.0	569,560

Table J.4: 5-d case: estimated errors for KDE and RMRP-KDE approximations.

	$\hat{d}_{KL}$	$\hat{L}_1$ error	Time (s)	Leaves
KDE ($n_K = 2,000$)	0.41	0.66	7,350–8,880	n/a
RMRP-KDE approximations				
$\overline{\psi} = 0.0001$	5.06	0.96	1.0	2,363
$\overline{\psi} = 0.00005$	4.85	0.91	2.3	4,639
$\overline{\psi} = 0.00001$	4.51	0.85	8.7	17,759
$\overline{\psi} = 0.000005$	4.49	0.84	17.2	31,335
$\overline{\psi} = 0.000001$	3.33	0.76	66.1	133,493
$\overline{\psi} = 0.0000005$	3.31	0.75	131.0	237,561
$\overline{\psi} = 0.0000001$	3.54	0.74	470.0	895,012

J.2 Estimated errors in estimating Density II using a KDE and an averaged RMRP histogram

Tables J.1, J.2, J.3, and J.4 show the full results discussed in Section 7.6. The tables show a summary of results for $\hat{d}_{KL}$ and $\hat{L}_1$ with true density Density II (see Appendix B) for $d = 2, 3, 4$ and 5. $\hat{d}_{KL}$ and $\hat{L}_1$ are shown for a KDE (using $n_K = 2,000$ sample points from the true distribution) and also for an RMRP estimate formed by averaging 100 SRP samples taken after burn-in from a Markov chain (thin-out 100), using the method described in Chapter 6. The results discussed in Section 6.7.2 and Appendix H suggest that a higher thin-out value would have increased the number of leaves in the average (and the time taken to make the estimate), but that the estimated errors would have been very close to those shown here.

The values shown for $\hat{d}_{KL}$ and $\hat{L}_1$ for the averaged histogram in Tables J.1, J.2, J.3 and J.4 are averages over 10 replications of the process with different sample data and different pseudo-random number sequences used for each replication. The minimum and maximum over the 10 replications are shown for both time taken and leaves for the averaged histogram RMRPs, except for the results for $n = 100,000$ in Table J.8 where time constraints meant that only three replications could be completed. The timings were recorded when other processes running on the same machine and can only be taken as a general indication of the time required to make these estimates.

Table J.5: 2-d case: estimated errors for KDE and averaged SRP histogram RMRP.

	$\hat{d}_{KL}$	$\hat{L}_1$ error	Time (s)		Leaves	
			min.	max.	min.	max.
KDE ($n_K = 2,000$)	0.04	0.20	5,000	7,200	*n/a*	
Averaged RMRP histogram						
$n = 5,000$	0.09	0.26	1	5	515	665
$n = 10,000$	0.06	0.22	2	13	811	902
$n = 50,000$	0.03	0.15	15	2,168	1,546	1,719
$n = 100,000$	0.02	0.12	49	420	2,000	2,236

Table J.6: 3-d case: estimated errors for KDE and averaged SRP histogram RMRP.

	$\hat{d}_{KL}$	$\hat{L}_1$ error	Time (s)		Leaves	
			min.	max.	min.	max.
KDE ($n_K = 2,000$)	0.13	0.35	5,600	7,200	*n/a*	
Averaged RMRP histogram						
$n = 5,000$	0.32	0.47	6	229	1,069	1,178
$n = 10,000$	0.24	0.41	21	451	1,573	1,718
$n = 50,000$	0.12	0.30	295	27,832	3,507	3,783
$n = 100,000$	0.09	0.26	1,694	49,199	5,150	5,513

Table J.7: 4-d case: estimated errors for KDE and averaged SRP histogram RMRP.

	$\hat{d}_{KL}$	$\hat{L}_1$ error	Time (s)		Leaves	
			min.	max.	min.	max.
KDE ($n_K = 2,000$)	0.25	0.51	7,200	8,050	*n/a*	
Averaged RMRP histogram						
$n = 5,000$	0.70	0.71	36	501	1,559	1,639
$n = 10,000$	0.57	0.64	160	1,754	2,381	2,531
$n = 50,000$	0.32	0.47	2,524	53,190	6,241	6,570
$n = 100,000$	0.25	0.42	10,382	82,684	9,431	9,775

Table J.8: 5-d case: estimated errors for KDE and averaged SRP histogram RMRP.

	$\hat{d}_{KL}$	$\hat{L}_1$ error	Time (s)		Leaves	
			min.	max.	min.	max.
KDE ($n_K = 2,000$)	0.41	0.66	7,350	8,880	*n/a*	
Averaged RMRP histogram						
$n = 5,000$	1.23	0.93	118	1,449	2,019	2,148
$n = 10,000$	1.02	0.85	858	5,470	3,093	3,334
$n = 50,000$	0.65	0.67	28,841	277,071	9,342	9,803
$n = 100,000$	0.53	0.60	24,244	399,016	15,160	15,563

Appendix K

Prior limits for RPABC

K.1 Introduction

One of the challenges of regular paving approximate Bayesian computation (RPABC), compared with adaptive approximate Bayesian computation (AABC), is identifying a suitable support for the prior. A large prior support is theoretically desirable but the computational efficiency of the method is affected by the size of the joint (parameter and summary statistic) sample space, and the parameter space is determined by the support of the prior. The efficiency of AABC methods is less sensitive to a wide prior support because parameter values are only drawn from the prior itself in the first iteration. In subsequent iterations parameter values are drawn from those which have already been evaluated as capable of generating at least one summary statistic reasonably close to the summary statistic of the observed data.

This appendix describes the heuristic initial step to establish a suitable support for the prior using the observed data that was added to the implementation of RPABC used in this thesis. The aim of this initial step is to create a modified version of a single, preliminary, AABC iteration, using the observed data to guide the range of parameter values from which to sample next.

K.1.1 Setting the support of the prior using the observed data

A tentative, wide, prior support is identified first. This gives the preliminary parameter space. This parameter space is overlaid by a grid of points. In the implementation used in this thesis the grid points are created so that there are 11 equally spaced unique values on each coordinate of the parameter space. For example, if the parameter space $\Theta = [0, 1] \in \mathbb{R}$ then the grid points would be 0.0, 0.1, 0.2, ..., 0.9, 1.0. If the parameter space
$\Theta = [0, 1] \times [10, 20] \in \mathbb{R}^2$, the grid of points would be

$$\begin{array}{cccc} (0.0, 10.0) & (0.1, 10.0) & \cdots & (1.0, 10.0) \\ (0.0, 11.0) & (0.1, 11.0) & \cdots & (1.0, 11.0) \\ \vdots & \vdots & \ddots & \vdots \\ (0.0, 19.0) & (0.1, 19.0) & \cdots & (1.0, 19.0) \\ (0.0, 20.0) & (0.1, 20.0) & \cdots & (1.0, 20.0) \end{array}$$

A sample of 100 summary statistics is drawn from $f(t \mid \theta')$ for each θ' in the grid of points. The Euclidian distance from each summary statistic thus simulated to each of the observed data summary statistics $t_1^\star, \ldots t_{n_{obs}}^\star$ is calculated and the collection of all such distances is used to find the 20^{th} percentile interval over the collection.

This interval is then compared with the distances to $t_1^\star, \ldots t_{n_{obs}}^\star$ for each of the 100 summary statistics simulated for each grid point θ'. Any θ' where *none* of the $100n_{obs}$ such distances is within the 20^{th} percentile interval calculated over all θ' grid values is discarded. At the end of the process the retained grid points are those which, in 100 attempts, simulated at least one piece of data with summary statistic relatively close to least one of the observed data summary statistics.

The lower limit of the prior support on each coordinate of the parameter space is then set to be the maximum of the lower limit of the tentative support on that coordinate and the smallest value of the θ' grid points retained on that coordinate less an additional 'safety margin' equal to the space between adjacent grid points on that coordinate. Similarly the upper limit of the prior support is set to be the minimum of the upper limit of the tentative support on that coordinate and the largest value of the θ' grid points retained plus a similar safety margin. Thus the prior support that results from the initial step will not be larger than the tentative support, but may be smaller.

For example, continuing the case of the parameter space $\Theta = [0, 1] \times [10, 20] \in \mathbb{R}^2$, suppose that the retained grid points are

$$\{(0.0, 15), (0.0, 16), (0.0, 17), (0.1, 9), (0.1, 10), (0.1, 8), (0.2, 5), (0.2, 6)\} \ ,$$

then the revised prior support will be $\Theta' = [0.0, 0.3] \times [4, 18]$.

K.1.2 Discussion

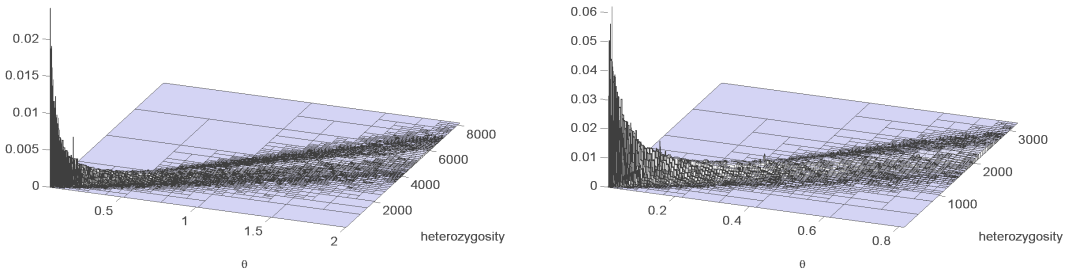
The specification of the grid, the number of samples of data drawn for each grid point, and the percentile interval used could all be adjusted to give a more or less thoroughly-investigated prior support. If the computational cost of drawing individual samples from $f(t \mid \theta')$ is very high, it might be desirable to use a more widely spaced grid of points, and possibly a larger interval, but a smaller number of simulations from $f(t \mid \theta')$.

Many other methods could also be used to achieve an equivalent end. The aim is to create a conservative guess at an appropriate prior support, guided by the observed data and the model.

The effect on the joint density of reducing the prior support for the example in Section 9.6 is illustrated in Figure K.1. Figure K.1(a) shows the PCF estimate of the joint density ($n = 1,000,000$ samples) with the preliminary prior support [0.020, 2.000]; Figure K.1(a) shows the PCF estimate of the joint density ($n = 500,000$ samples) with the adjusted prior support [0.020, 0.814]. The larger sample size for the estimate with the wider prior support compensates for the increased size of the joint sample space.

The larger sample space and increased sample size mean that it takes longer to form the MCMC RMRP density estimate. In both cases the MCMC process used three chains and a

total of 100 SRP samples were collected from all three chains after convergence was diagnosed, with a thin-out of 1,000. The time taken to simulate 1,000,000 samples and then form the RMRP for Figure K.1(a) was about 5–6 hours. The time taken to simulate 500,000 samples and obtain the RMRP for Figure K.1(b) was about 80 minutes. The initial step described in Section K.1.1 itself took about 12 minutes for Figure K.1(b). The benefits of using the initial step are exaggerated in this example because of the very large number of simulations from the model used. The initial step itself also involves a considerable number of simulations from the model, although, as described above, this could be reduced if required. However, if data can be simulated from the model quite quickly then the initial step costs relatively little, and if it is computationally costly to simulate from the model then the expenditure of some additional simulations in the initial step may be repaid by being able to concentrate the simulations used to form the joint density estimate on most relevant parameter values. In this particular population genetics example the larger values of θ removed from the prior support by the initial step also reduced the average cost of each simulation required for the joint density.



(a) Prior Uniform(0.020, 2.000). (b) Prior Uniform(0.020, 0.814).

Figure K.1: The effect on the joint density of reducing the prior support for the example in Section 9.6.

Appendix L

Data for RPABC examples

L.1 Data for multivariate Normal mixture examples

The first 10 artificial observations of x for the univariate Normal mixture example and of x_1, x_2 for the bivariate Normal mixture example described in Chapter 9 are listed below.

x	$\boldsymbol{x}_1$	x_2
0.313	0.313	1.813
0.011	0.104	0.020
−0.110	−0.223	−0.209
−0.068	−0.008	−0.098
0.333	−0.102	0.144
−0.081	0.246	0.051
0.053	0.011	−0.042
−0.119	2.671	−0.885
0.033	0.666	−1.507
0.051	−0.788	0.684

The average for the first 10 observations of x (to 3 decimal places) was 0.042. The average for all 50 observations was -0.118.

The averages for the first 10 observations of x_1 and x_2 (to 3 decimal places) were $x_1 = 0.289$, $x_2 = -0.003$.

The averages for all 50 observations of x_1 and x_2 (to 3 decimal places) were $x_1 = 0.117$ and $x_2 = -0.080$, respectively.

L.2 Data for population genetics examples

The segregating sites and heterozygosity summary statistics of the artificial observed data for the first 10 individual loci used for the population genetics examples described in Chapter 9 are listed below.

Segregating sites	Heterozygosity
2325	703.578
2217	613.200
2176	555.556
1783	497.533
2458	605.422
2036	557.711
2316	575.422
1395	368.622
2039	609.311
1740	565.711

The average segregating sites statistic over the first 10 loci was 2048.5. The average heterozygosity statistic over the first 10 loci was 565.207.

The average segregating sites statistic over all 50 loci was 1989.8. The average heterozygosity statistic over all 50 loci was 545.800.

References

Baltrunas, L., Mazeika, A., and Bohlen, M. (2006), "Multi-Dimensional Histograms with Tight Bounds for the Error," in *Proceedings of the 10th International Database Engineering and Applications Symposium*, Washington, D.C.: IEEE Computer Society, pp. 105–112.

Bashtannyk, D. M. and Hyndman, R. J. (2001), "Bandwidth Selection for Kernel Conditional Density Estimation," *Computational Statistics & Data Analysis*, 36, 279–298.

Beaumont, M. A. (2008), "Joint Determination of Topology, Divergence Time, and Immigration in Population Trees," in *Simulation, Genetics, and Human Prehistory*, eds. Matsumura, S., Forster, P., and Renfrew, C., Cambridge, United Kingdom: McDonald Institute for Archealogical Research, pp. 135–154.

— (2010), "Approximate Bayesian Computation in Evolution and Ecology," *Annual Review of Ecology, Evolution, and Systematics*, 41, 379–406.

Beaumont, M. A., Cornuet, J.-M., Marin, J.-M., and Robert, C. P. (2009), "Adaptive Approximate Bayesian Computation," *Biometrika*, 96, 983–990.

Beaumont, M. A., Zhang, W., and Balding, D. J. (2002), "Approximate Bayesian Computation in Population Genetics," *Genetics*, 162, 2025–2035.

Bertorelle, G., Benazzo, A., and Mona, S. (2010), "ABC as a Flexible Framework to Estimate Demography over Space and Time: Come Cons, Many Pros," *Molecular Ecology*, 19, 2609–2625.

Biau, G., Cadre, B., Devroye, L., and Györfi, L. (2007), "Strongly Consistent Model Selection for Densities," *Test*, 17, 531–545.

Birgé, L. and Rozenholc, Y. (2006), "How Many Bins Should be put in a Regular Histogram," *ESAIM: Probability and Statistics*, 10, 24–25.

Blum, M. G. B. and François, O. (2010), "Non-Linear Regression models for Approximate Bayesian Computation," *Statistics and Computing*, 20, 63–73.

Bolstad, W. M. (2010), *Understanding Computational Bayesian Statistics*, Hoboken, NJ: Wiley.

Bowman, A. W. and Azzalini, A. (1997), *Applied Smoothing Techniques for Data Analysis*, Oxford, United Kingdom: Clarendon Press.

Brooks, S. and Gelman, A. (1998), "General Methods for Monitoring Convergence of Iterative Simulations," *Journal of Computational and Graphical Statistics*, 7, 434–455.

Brooks, S. P. (1998), "Markov Chain Monte Carlo Method and its Application," *Journal of the Royal Statistical Society. Series D (The Statistician)*, 47, 69–100.

Cameron, E. and Pettitt, A. N. (2012), "Approximate Bayesian Computation for Astronomical Model Analysis: A Case Study in Galaxy Demographics and Morphological Transformation at High Redshift," *Monthly Notices of the Royal Astronomical Society*, 425, 44–65.

Cappé, O., Guillin, A., Marin, J. M., and Robert, C. P. (2004), "Population Monte Carlo," *Journal of Computational and Graphical Statistics*, 13, 90–929.

Castellan, G. (1999), "Modified Akaike's Criterion for Histogram Density Estimation," Tech. rep., Université Paris-Sud, Orsay.

Cheng, K. F., Chu, C. K., and Lin, D. K. J. (2006), "Quick Multivariate Kernel Density Estimation for Massive Data Sets," *Applied Stochastic Models in Busines and Industry*, 22, 533–546.

Cornuet, J.-M., Santos, F., Beaumont, M. A., Robert, C. P., Marin, J.-M., Balding, D. J., Guillemaud, T., and Estoup, A. (2008), "Inferring Population History with DIY ABC: A User-Friendly Approach to Approximate Bayesian Computation," *Bioinformatics*, 24, 2713–2719.

Cowles, M. and Carlin, B. (1996), "Markov Chain Monte Carlo Convergence Diagnostics: A Comparative Review," *Journal of the American Statistical Association*, 91, 883–904.

Csilléry, K., Blum, M. G. B., Gaggiotti, O. E., and François, O. (2010), "Approximate Bayesian Computation (ABC) in Practice," *Trends in Ecology and Evolution*, 25, 410–418.

Devroye, L. and Györfi, L. (1992), *Nonparametric Density Estimation: The L_1 View*, New York: Wiley.

Devroye, L., Györfi, L., and Lugosi, G. (1996), *A Probabilistic Theory of Pattern Recognition*, New York: Springer-Verlag.

Devroye, L. and Lugosi, G. (2001), *Combinatorial Methods in Density Estimation*, New York: Springer-Verlag.

— (2004), "Bin Width Selection in Multivariate Histograms by the Combinatorial Method," *TEST*, 13, 129–145.

Douc, R., Guillan, A., Marin, J.-M., and Robert, C. P. (2007), "Convergence of Adaptive Mixtures of Importance Sampling Schemes," *Annals of Statistics*, 35, 420–448.

Duong, T. and Hazelton, M. (2003), "Plug-In Bandwidth Matrices for Bivariate Kernel Density Estimation," *Journal of Nonparametric Statistics*, 15, 17–30.

Duong, T. and Hazelton, M. L. (2005), "Convergence Rrates for Unconstrained Bandwidth Matrix Selectors in Multivariate Kernel Density Estimation," *Journal of Multivariate Analysis*, 93, 417–433.

Excoffier, L. and Lischer, H. E. L. (2010), "Arlequin Suite ver 3.5: A New Series of Programs to Perform Population Genetics Analyses under Linux and Windows," *Molecular Ecology Resources*, 10, 5640–567.

Fan, J. and Marron, J. S. (1994), "Fast Implementations of Nonparametric Curve Estimators," *Journal of Computational and Graphical Statistics*, 3, 35–56.

Fan, J. and Yim, T. H. (2004), "A Crossvalidation Method for Estimating Conditional Densities," *Biometrika*, 91, 819–834.

Fearnhead, P. and Prangle, D. (2012), "Constructing Summary Statistics for Approximate Bayesian Computation: Semi-Automatic ABC," *Journal of the Royal Statistical Society. Series B (Methodological)*, 74, 419–474.

Filippone, M. and Sanguinetti, G. (2011), "Approximate Inference of the Bandwidth in Multivariate Kernel Density Estimation," *Computational Statistics & Data Analysis*, 55, 3104–3122.

Fitzpatrick, B. M. (2009), "Power and Sample Size for Nested Analysis of Molecular Variance," *Molecular Ecology*, 18, 3961–3966.

Gelman, A. and Rubin, D. B. (1992), "Inference from Iterative Simulation Using Multiple Sequences," *Statistical Science*, 7, 457–472.

Gessaman, M. P. (1970), "A Consistent Nonparametric Multivariate Density Estimator Based on Statistically Equivalent Blocks," *The Annals of Mathematical Statistics*, 41, 1344–1346.

Gray, A. G. and Moore, A. W. (2003a), "Nonparametric Density Estimation: Towards Computational Tractability," in *SIAM International Conference on Data Mining*, SIAM, pp. 203–211.

— (2003b), "Rapid Evaluation of Multiple Density Models," in *Proceedings of the Ninth Conference on Artificial Intelligence and Statistics (AISTATS)*.

Guha, S. and Koudas, N. (2002), "Approximating a Data Stream for Querying and Estimation: Algorithms and Performance Evaluation," in *Proceedings of the 18th International Conference on Data Engineering (ICDE'02)*, pp. 567–576.

Guillemaud, T., Beaumont, M. A., Ciosi, M., Cornuet, J.-M., and Estoup, A. (2009), "Inferring Introduction Routes of Invasive Species Using Approximate Bayesian Computation on Microsatellite Data," *Heredity*, 104, 88–99.

Györfi, L. and Kohler, M. (2007), "Nonparametric Estimation of Conditional Distributions," *IEEE Transactions on Information Theory*, 53, 1872–1879.

Hall, P. (1987), "On Kullback-Leibler Loss and Density Estimation," *The Annals of Statistics*, 15, 1491–1519.

Hall, P., Racine, J., and Li, Q. (2004), "Cross-Validation and the Estimation of Conditional Probability Densities," *Journal of the American Statistical Association*, 99, 1015–1026.

Harlow, J., Sainudiin, R., and Tucker, W. (2012), "Mapped Regular Pavings," *Reliable Computing*, 16, 252–282.

Hartig, F., Calabrese, J. M., Reineking, B., Wiegand, T., and Huth, A. (2011), "Statistical Inference for Stochastic Simulation Models — Theory and Application," *Ecology Letters*, 14, 816–827.

Hudson, R. R. (2002), "Generating Samples Under a Wright-Fisher Neutral Model of Genetic Variation," *Bioinformatics*, 18, 337–338.

Hudson, R. R., Boos, D. D., and Kaplan, N. L. (1992), "A Statistical Test for Detecting Geographic Subdivision," *Molecular Biology and Evolution*, 9, 138–151.

Jaulin, L., Kieffer, M., Didrit, O., and Walter, E. (2001), *Applied Interval Analysis with Examples in Parameter and State Estimation, Robust Control and Robotics*, London: Springer-Verlag.

Johnson, V. E. (1996), "Studying Convergence of Markov Chain Monte Carlo Algorithms Using Coupled Sample Paths," *Journal of the American Statistical Association*, 91, 154–166.

Jović, M., Stejić, Z., Seidl, T., and Assent, I. (2004), "Image Clustering and Retrieval Combining Fixed/Adaptive-Binned Histograms and Various Distance Functions," in *Proceedings of the 2004 IEEE Conference on Cybernetics and Intelligent Systems*, vol. 1, pp. 301–306.

Karunamuni, R. and Alberts, T. (2005), "On Boundary Correction in Kernel Density Estimation," *Statistical Methodology*, 2, 191–212.

Kieffer, M., Jaulin, L., Braems, I., and Walter, E. (2001), "Guaranteed Set Computation with Subpavings," in *Scientific Computing, Validated Numerics, Interval Methods, Proceedings of SCAN 2000*, eds. Kraemer, W. and Gudenberg, J., New York: Kluwer Academic Publishers, pp. 167–178.

Kimura, M. (1969), "The Number of Heterozygous Nucleotide Sites Maintained in a Finite Population due to Steady Flux of Mutations," *Genetics*, 61, 893–903.

Klemelä, J. (2009), *Smoothing of Multivariate Data: Density Estimation and Visualization*, Chichester, United Kingdom: Wiley.

Knuth, D. E. (2006a), *The Art of Computer Programming: Generating All Trees; History of Combinatorial Generation*, vol. 4, fascile 4, Upper Saddle River, NJ: Addison-Wesley.

Knuth, K. H. (2006b), "Optimal Data-Based Binning for Histograms," *arXiv preprint physics/0605197*, 1–22.

Kruse, R. L. (1987), *Data Structures and Program Design*, Englewood Cliffs, NJ: Prentice-Hall, chap. 8, 2nd ed., pp. 273–317.

Kuhner, M. K. (2006), "LAMARC 2.0 : Maximum Likelihood and Bayesian Estimation of Population Parameters," *Bioinformatics*, 22, 768–770.

Kuhner, M. K. and Smith, L. P. (2007), "Comparing Likelihood and Bayesian Coalescent Estimation of Population Parameters," *Genetics*, 175, 155–165.

Kuhner, M. K., Yamato, J., and Felsenstein, J. (2000), "Maximum Likelihood Estimation of Recombination Rates from Population Data," *Genetics*, 156, 1393–1401.

Lee, D. and Gray, A. (2009), "Fast High-Dimensional Kernel Summations Using the Monte Carlo Multipole Method," in *Advances in Neural Information Processing Systems (NIPS), 21 (2008)*, MIT Press, pp. 929–936.

Lee, D., Gray, A., and Moore, A. (2006), "Dual-Tree Fast Gauss Transforms," in *Advances in Neural Information Processing Systems (NIPS), 18 (2005)*, MIT Press, pp. 747–754.

Lee, D. S. (2008), "Exact Markov Chain Monte Carlo Algorithms and Their Applications in Probabilistic Data Analysis and Inference," in *Intelligent Data Analysis: Developing New Methodologies Through Pattern Discovery and Recovery*, ed. Wang, H.-F., Hershey, PA: IGI Publishing, pp. 161–183.

Leuenberger, C. and Wegmann, D. (2010), "Bayesian Computation and Model Selection Without Likelihoods," *Genetics*, 184, 243–252.

Levin, D. A., Peres, Y., and Wilmer, E. L. (2009), *Markov Chains and Mixing Times*, Providence, RI: American Mathematical Society.

Link, W. A. and Eaton, M. J. (2012), "On Thinning of Chains in MCMC," *Methods in Ecology and Evolution*, 3, 112–115.

Loftsgaarden, D. O. and Quesenberry, C. P. (1965), "A Nonparametric Estimate of a Multivariate Density Function," *The Annals of Mathematical Statistics*, 36, 1049–1051.

Lugosi, G. and Nobel, A. (1996), "Consistency of Data-Driven Histogram Methods for Density Estimation and Classification," *The Annals of Statistics*, 24, 687–706.

Marjoram, P. and Tavaré, S. (2006), "Modern Computational Approaches for Analysing Molecular Genetic Variation Data," *Nature Reviews Genetics*, 7, 759–770.

Massert, P. (2007), *Concentration Inequalities and Model Selection: Ecole d'Eté de Probabilités de Saint-Flour XXXIII — 2003*, Berlin, Germany: Springer-Verlag.

Meier, J. (2008), *Groups, Graphs and Trees: An Introduction to the Geometry of Infinite Groups*, Cambridge, United Kingdom: Cambridge University Press.

Müller, E., Assent, I., Krieger, R., Günnemann, S., and Seidl, T. (2009), "DensEst : Density Estimation for Data Mining in High Dimensional Spaces," in *SIAM International Conference on Data Mining*, SIAM, pp. 175–186.

Niederreiter, H. (1992), *Random Number Generation and Quasi-Monte Carlo methods*, Philadelphia: Society for Industrial and Applied Mathematics.

O'Hagan, A. and Forster, J. (2004), *Kendall's Advanced Theory of Statistics*, vol. 2B (Bayesian Inference), London: Arnold, 2nd ed.

P754, I. T. (1985), *ANSI/IEEE 754-1985, Standard for Binary Floating-Point Arithmetic*, IEEE, New York.

Pawitan, Y. (2001), *In All Likelihood: Statistical Modelling and Inference Using Likelihood*, Oxford, United Kingdom: Clarendon Press.

Propp, J. G. and Wilson, D. B. (1996), "Exact Sampling with Coupled Markov Chains and Applications to Statistical Mechanics," *Random Structures and Algorithms*, 9, 223–252.

Ray, N., Wegmann, D., Fagundes, N. J. R., Wang, S., Ruiz-Linares, A., and Excoffier, L. (2010), "A Statistical Evaluation of Models for the Initial Settlement of the American Continent Emphasizes the Importance of Gene Flow with Asia," *Molecular Biology and Evolution*, 27, 337–345.

Raykar, V. C., Duraiswami, R., and Zhao, L. H. (2010), "Fast Computation of Kernel Estimators," *Journal of Computational and Graphical Statistics*, 19, 205–220.

Raymond, M. and Rousset, F. (1995), "An Exact Test for Population Differentiation," *Evolution*, 49, 1280–1283.

Rissanen, J., Speed, T., and Yu, B. (1992), "Density Estimation by Stochastic Complexity," *IEEE Transactions on Information Theory*, 38, 315–323.

Robert, C. P., Cornuet, J.-M., Marin, J.-M., and Pillai, N. S. (2011), "Lack of Confidence in Approximate Bayesian Computation Model Choice," *Proceedings of the National Academy of Sciences of the United States of America*, 108, 15112–15117.

Rozenholc, Y., Mildenberger, T., and Gather, U. (2009), "Constructing Irregular Histograms by Penalized Likelihood," Tech. rep., Technische Universität Dortmund, Sonderforschungsbereich 475.

Rübel, O., Prabhat, Wu, K., Childs, H., Meredith, J., Geddes, C. G. R., Cormier-Michel, E., Ahern, S., Weber, G. H., Messmer, P., Hagen, H., Hamann, B., and Bethel, E. W. (2008), "High Performance Multivariate Visual Data Exploration for Extremely Large Data," in *International Conference for High Performance Computing, Networking, Storage and Analysis (SC 2008)*, pp. 1–12.

Sain, S. R. (2002), "Multivariate Locally Adaptive Density Estimation," *Computational Statistics & Data Analysis*, 39, 165–186.

Sainudiin, R., Teng, G., Harlow, J., and Lee, D. S. (2013), "Posterior Expectation of Regularly Paved Random Histograms," *ACM Transactions on Modeling and Computer Simulation*, 23, (Accepted for publication).

Samet, H. (1990), *The Design and Analysis of Spatial Data Structures*, Boston: Addison-Wesley Longman.

— (2006), *Foundations of Multidimensional and Metric Data Structures*, San Francisco: Morgan Kaufman.

Scott, D. W. (1992), *Multivariate Density Estimation*, New York: Wiley.

Scott, D. W. and Sain, S. R. (2005), "Multidimensional Density Estimation," in *Handbook of Statistics*, eds. Rao, C. R., Wegman, E. J., and Solka, J. L., Amsterdam, The Netherlands: Elsevier, vol. 24, chap. 9, pp. 229–262.

Sheather, S. J. (2004), "Density Estimation," *Statistical Science*, 19, 588–597.

Silverman, B. W. (1986), *Density Estimation for Statistics and Data Analysis*, London: Chapman and Hall.

Sisson, S. A., Fan, Y., and Tanaka, M. M. (2007), "Sequential Monte Carlo Without Likelihoods," *Proceedings of the National Academy of Sciences of the United States of America*, 104, 1760–1765.

Sorensen, D. and Gianola, D. (2002), *Likelihood, Bayesian, and MCMC Methods in Quantitative Genetics*, New York: Springer.

Stanley, R. P. (1999), *Enumerative Combinatorics, Vol. 2*, Cambridge, United Kingdom: Cambridge University Press.

Stone, C. J. (1985), "An Asymptotically Optimal Histogram Selection Rule," in *Proceedings of the Berkeley Conference in Honor of Jerzy Neyman and Jack Kiefer, Vol. II*, Belmont, CA: Wadsworth, pp. 513–520.

Taylor, C. C. (1987), "Akaike's Information Criterion and the Histogram," *Biometrika*, 74, 636–639.

Teng, G. (2013), "Statistical Regular Pavings and Their Applications," Ph.D. thesis, University of Canterbury.

Thornton, K. R. (2009), "Automating Approximate Bayesian Computation by Local Linear Regression," *BMC Genetics*, 10, 35.

Tian, T., Xu, S., Gao, J., and Burrage, K. (2007), "Simulated Maximum Likelihood Method for Estimating Kinetic Rates in Gene Expression," *Bioinformatics*, 23, 84–91.

Tierney, L. (1994), "Markov Chains for Exploring Posterior Distributions," *The Annals of Statistics*, 22, 1701–1728.

Tukey, J. W. (1947), "Non-Parametric Estimation II. Statistically Equivalent Blocks and Tolerance Regions — The Continuous Case," *The Annals of Mathematical Statistics*, 18, 529–539.

Wakeley, J. (2009), *Coalescent Theory: An Introduction*, Greenwich Village, CO: Roberts and Company.

Wand, M. P. (1994), "Fast Computation of Multivariate Kernel Estimators," *Journal of Computational and Graphical Statistics*, 3, 433–445.

— (1997), "Data-Based Choice of Histogram Bin Width," *The American Statistician*, 51, 59–64.

Wand, M. P. and Jones, M. C. (1994), "Multivariate Plug-In Bandwidth Selection," *Computational Statistics*, 9, 97–116.

Wasserman, L. (2003), *All of Statistics: A Concise Course in Statistical Inference*, New York: Springer.

— (2007), *All of Nonparametric Statistics*, New York: Springer.

Wegmann, D., Leuenberger, C., Neuenschwander, S., and Excoffier, L. (2010), "ABCtoolbox: A Versatile Toolkit for Approximate Bayesian Computations," *BMC Bioinformatics*, 11, 116.

Whittle, P. (1958), "On the Smoothing of Probability Density Functions," *Journal of the Royal Statistical Society . Series B (Methodological)*, 20, 334–343.

Wood, S. N. (2010), "Statistical Inference for Noisy Nonlinear Ecological Dynamic Systems," *Nature*, 466, 1102–1104.

Zhang, X., King, M. L., and Hyndman, R. J. (2006), "A Bayesian Approach to Bandwidth Selection for Multivariate Kernel Density Estimation," *Computational Statistics & Data Analysis*, 50, 3009–3031.